51CTO.com
技术成就梦想

我 们 只 谈 开 发

开发月刊

Development Monthly

2013年08月
总第029期

JavaScript游戏开发实例指南

在浏览器中攻破JavaScript到底有多轻松？

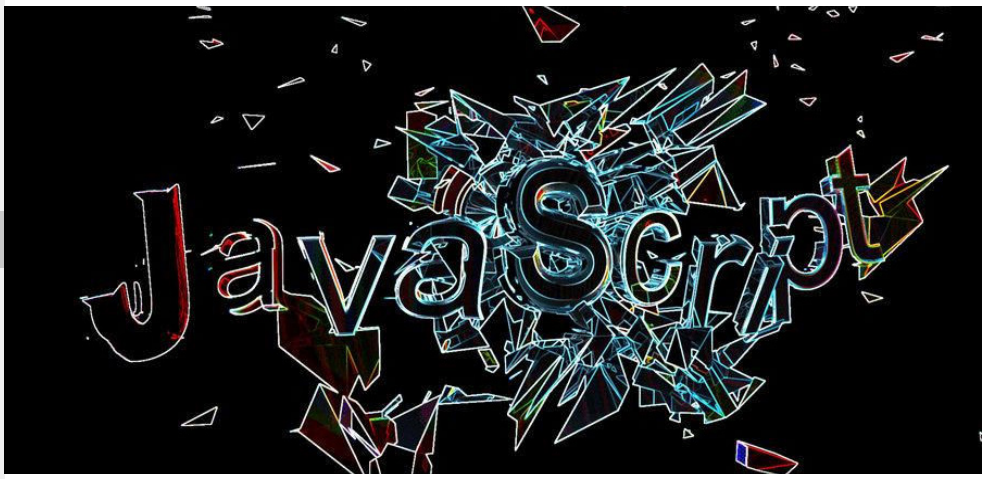


专访W3C中国区负责人李安琪

而在去年，由于HTML5的标准制定者WHATWG工作组和W3C最终选择了分道扬镳，给W3C组织很大的打击。但是，坚持照传统编号的版本控制系统构建一个静态快照版标准的W3C与致力于为HTML5制定一个不断进化的动态标准的WHAT工作……

51CTO.com
技术成就梦想

开发月刊





出版方：

北京无忧创想信息技术有限公司

责任编辑：

林师授　陈四芳

封面设计：

苍旭

联系方法：

邮箱：linss@51cto.com

电话：010-68476606-8123

出版日期：2013年08月29日

欢迎来稿，请发送邮件至

linss@51cto.com

主编的话

本期《开发月刊》的主推内容是JavaScript相关技术，以原创译文为主，把国外最新，最热的技术展现给大家。

不可否认，自从90年代中期发布以来，JavaScript已经成为最流行的Web开发语言之一。在去年的9月份，行业分析公司RedMonk就表示其已成为软件开发领域的顶级语言，这主要是由于JavaScript能够提供丰富的动态网页内容，以及其轻巧和易用的特性。

同时，开发月刊集合了热门的资源下载，最近热门技术专题以及相关开发频道每周重点推荐的内容。不管朋友们喜不喜欢我们推荐的文章，但是我们会一如既往的把开发最好的内容推荐给网友，与网友共同进步，成长。

祝好！

51CTO开发频道寄语

编程排行 Billboard

04 8月编程语言排行榜：C和Objective-C成受害者

专题报道 Special report

06 想追赶.Net的脚步？Java面前障碍重重
09 JavaScript游戏开发实例指南
13 HTML5 WebSockets node.js实例教程
15 在浏览器中攻破JavaScript到底有多轻松？

技术热点 Techlogy hot

18 只会编程的程序员没有前途
19 权国清：谈PHP发展现状
21 系统级编程语言性能大PK D语言成首选
23 用什么编程语言的程序员赚钱最多
26 我也谈谈.NET程序员工资低
28 保障IT生涯顺利晋升下一级别的14种方式
30 程序员：浮躁世界，我思考
32 Java程序员在写SQL程序时候常犯的10个错误
34 12 岁的孩子教你使用 Python 开发游戏
36 对九个超级程序员的采访
38 数据即代码,我和小伙伴们都惊呆了！
40 MOCO作者郑晔:谈MOCO框架技术和Java未来
42 专访W3C中国区负责人李安琪： HTML5标准进展与最佳实践

■ 编者按

在本月排行中，TIOBE增加了16个新的搜索引擎后，C和Objective-C成为最大的受害者，它们呈大幅下降趋势，分别下降了1.65% 和2.17%；而Go 语言则从42 名提升到26 名。

8月编程语言排行榜：C和Objective-C成受害者

为了进一步提升TIOBE指数排行的专业性，截止本月，我们已经增加了16个新的搜索引擎，这些搜索引擎绝大多数来自中国和美国。甚至应该和巴西的网站也都在应用。这些搜索引擎的列表请看这里。这些新的搜索引擎的介绍在下个月将全部完成。感兴趣的朋友可以在TIOBE排行指数的定义页面上找到。

在本月排行中，TIOBE增加了16个新的搜索引擎后，C和Objective-C成为最大的受害者，它们呈大幅下降趋势，分别下降了1.65% 和2.17%；而Go 语言则从42 名提升到26 名。LabVIEW者从100提升到49名，Openedge ABL从129名提升到57名。而Java又重新回到了自己的宝座。 以下是前20名榜单排行榜：

Position Aug 2013	Position Aug 2012	Delta in Position	Programming Language	Ratings Aug 2013	Delta Aug 2012	Status
1	2	↑	Java	15.978%	-0.37%	A
2	1	↓	C	15.974%	-2.96%	A
3	4	↑	C++	9.371%	+0.04%	A
4	3	↓	Objective-C	8.082%	-1.46%	A
5	6	↑	PHP	6.694%	+1.17%	A
6	5	↓	C#	6.117%	-0.47%	A
7	7	=	(Visual) Basic	3.873%	-1.46%	A
8	8	=	Python	3.603%	-0.27%	A
9	11	↑↑	JavaScript	2.093%	+0.73%	A
10	10	=	Ruby	2.067%	+0.38%	A
11	9	↓↓	Perl	2.041%	-0.23%	A
12	15	↑↑↑	Transact-SQL	1.393%	+0.54%	A
13	14	↑	Visual Basic .NET	1.320%	+0.44%	A
14	12	↓↓	Delphi/Object Pascal	0.918%	-0.09%	A-
15	20	↑↑↑↑↑	MATLAB	0.841%	+0.31%	A-
16	13	↓↓↓	Lisp	0.752%	-0.22%	A
17	19	↑↑	PL/SQL	0.751%	+0.14%	A
18	16	↓↓	Pascal	0.620%	-0.17%	A-
19	23	↑↑↑↑	Assembly	0.616%	+0.11%	B
20	22	↑↑	SAS	0.580%	+0.06%	B

LabVIEW介绍

与C和BASIC一样，LabVIEW也是通用的编程系统，有一个完成任何编程任务的庞大函数库。LabVIEW的函数库包括数据采集、GPIB、串口控制、数据分析、数据LabVIEW标志 LabVIEW标志

显示及数据存储，等等。LabVIEW也有传统的程序调试工具，如设置断点、以动画方式显示数据及其子程序（子VI）的结果、单步执行等等，便于程序的调试。

LabVIEW（Laboratory Virtual Instrument Engineering Workbench）是一种用图标代替文本行创建应用程序的图形化编程语言。传统文本编程语言根据语句和指令的先后顺序决定程序执行顺序，而LabVIEW 则采用数据流编程方式，程序框图中节点之间的数据流向决定了VI及函数的执行顺序。VI指虚拟仪器，是 LabVIEW的程序模块。

LabVIEW提供很多外观与传统仪器（如示波器、万用表）类似的控件，可用来方便地创建用户界面。用户界面在 LabVIEW 中被称为前面板。使用图标和连线，可以通过编程对前面板上的对象进行控制。这就是图形化源代码，又称G代码。LabVIEW 的图形化源代码在某种程度上类似于流程图，因此又被称作程序框图代码。

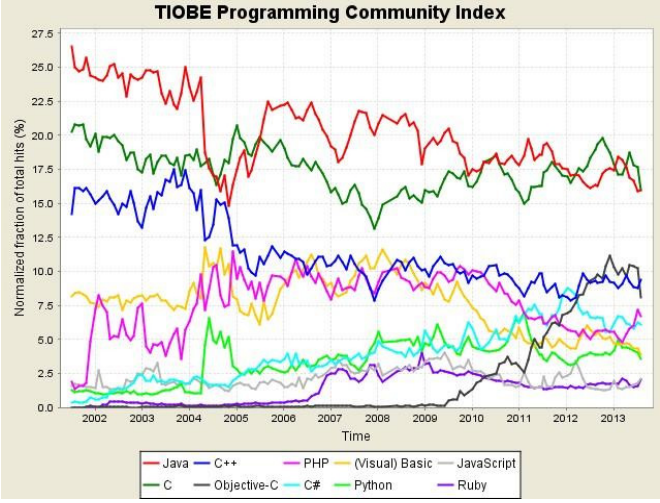
LabVIEW特点

特点尽可能采用了通用的硬件，各种仪器的差异主要是软件。

可充分发挥计算机的能力，有强大的数据处理功能，可以创造出功能更强的仪器。

用户可以根据自己的需要定义和制造各种仪器。

前10名编程语言走势图：



下面是第50到100的编程语言排名：

(Visual) FoxPro, 4th Dimension/4D, Algol, Alice, APL, Arc, Awk, bc, C shell, CFML, cg, Clean, Clojure, cT, Dart, Eiffel, Factor, Falcon, Forth, Icon, J, J#, JScript.NET, Ladder Logic, MAD, Magic, Modula-2, Monkey, Moto, NATURAL, NXT-G, OCaml, OpenCL, OpenEdge ABL, Oz, PILOT, PostScript, Pure Data, Q, REALbasic, Revolution, S, SPARK, SPSS, Squirrel, TOM, VBScript, VHDL, X10, Z shell

本文未完，请点击下面连接阅读全文：
http://developer.51cto.com/art/201308/406121.htm

20到50名语言排行

Position	Programming Language	Ratings
21	ABAP	0.567%
22	D	0.567%
23	F#	0.538%
24	Lua	0.531%
25	Logo	0.523%
26	Go	0.508%
27	COBOL	0.500%
28	Bash	0.491%
29	Ada	0.469%
30	Scheme	0.461%
31	Fortran	0.415%
32	R	0.395%
33	Scratch	0.354%
34	Haskell	0.349%
35	Erlang	0.344%
36	Prolog	0.334%
37	Scala	0.332%
38	ActionScript	0.309%
39	Smalltalk	0.297%
40	ML	0.249%
41	Emacs Lisp	0.230%
42	RPG (OS/400)	0.227%
43	ATLAS	0.223%
44	Groovy	0.211%
45	Inform	0.199%
46	Tcl	0.188%
47	Common Lisp	0.186%
48	ABC	0.183%
49	LabVIEW	0.182%
50	Io	0.181%

编者按

待到Java 8面世之时 .Net的进度时钟恐怕已经又走过了两到五年——届时微软做出的调整将使二者差距进一步拉大。就在几周之前，我详细介绍了Java 8中值得期待的几大主要功能。

想追赶.Net的脚步？Java面前障碍重重

---待到Java 8面世之时 .Net的进度时钟恐怕已经又走过了两到五年——届时微软做出的调整将使二者差距进一步拉大。

就在几周之前，我详细介绍了Java 8中值得期待的几大主要功能。不过当时我并没有提到.Net的新变化，事实上Java 8中的大部分（甚至全部）功能都能在.Net中找到。更夸张的是，不少将被推迟到Java 9中实现的功能也将在.Net中出现。

我并不赞成将一切功能盲目塞进Java语言的激进行为，不过我认为Java平台（相对于语言本身）确实应该在功能多样性方面下点功夫。在我看来，.Net技术堪称杰出，C#与.Net平台自Java 3时代就开始在各个方面迎头赶上。就个人而言，我对微软的操作系统非常抵触，而且很担心无法修复讨厌的bug（至少在理论上不行）。

两套平台、一个故事

很多朋友认为微软公司在提供较小安装基础与激发开发者拥护热情方面行动更快，这样的论断还算公正。我还记得上世纪九十年代与两千年初时，微软公司决定以几乎每周一次的速度变更数据库API，于是ODBC、RDO、ADO乃至OLEDB等等一下子涌到我们面前。然而随着.Net的出现，微软的研发强度达到了临界值，后续而来的是更凶猛、更频繁的发展进程。

然而Java为什么会落后如此之多？在Java出现的早期，其发展速度同样令人赞叹。从Java 1.0.2到Java 1.1，我们仅在一年之间就迎来了众多根本性（通常也意味着存在兼容性问题）改变。其后，从1.1版本到1.2版本用了一年半时间，之后的1.22——一个看似小更新、实为大升级的版本——仅在七个月后就火热出炉。

短短十个月后，里程碑式的Java 1.3版本整装待发，这也是第一个考虑在服务器端加入垃圾收集功能的版本。

Java 1.4给我们带来了NIO（即网络接口对象）与正则表达式，与前代版本相隔不到两年。Java 1.4.2则在多核环境中实现了垃圾收集功能（虽然还不太稳定），开发周期为一年。接下来是Java 1.5，这个开发周期超过一年的新版本将并发一致性GC引入生产流程，并且加入了其它一些重要的并发及NIO功能。

Java 1.6将关注重点放在性能节约方面，虽然效果还算显著，但其改进幅度仍然无法与1.5版本相提并论、更遑论用去了无数开发者两年的等待时间。Java 1.7是自1.4.2以来第一个针对底层虚拟机技术（G1 collector）做出大幅改动的新版本，利用invokedynamic指令帮助我们在JVM环境下更好地与其它语言对接。尽管属于大版本升级，但五年的更新周期无疑标志着Java的迭代步伐已经明显放缓。

Sample features in Java and .Net and their release dates

Java功能	.Net或C#功能	Java版本及日期	.Net 或 C# 版本及日期
java.util.concurrentFuture/ ForkJoinPool / java.util.stream	任务并行库	Java 5 / 2004年9月30日 Java 7/2011年7月28日 Java 8 / 2014年4月	.Net 4.0 /2010年4月12日
Lambda表达式	Lambda 表达式	Java 8/2014年4月	.Net 3.5 /2007年11月19日
switch语句中的字符串	C# switch	Java 7 / 2011年7月28日	.Net 1.1 / 2003年4月24日
泛型	泛型	Java 5 / 2004年9月30日	.Net 2.0 / 2005年11月7日
NIO	异步I/O	Java 1.4 / 2002年2月6日	.Net 2.0 / 2005年11月7日
Jigsaw	程序集与应用程序域	Java 9 / ?	.Net 1.1 / 2003年2月3日 (在后续版本中持续改进)

进展为何如何缓慢?

我们可以这样来简单解释Java的逐渐落后：Sun本来就不是一家运转状况良好的企业。Java诞生之初互联网正迅速兴起，Sun公司也将运营重点放在了销售Sparc及相关产品方面。

与此同时，英特尔与AMD产品的价格逐步下降，Sparc的价格却未作调整。尽管T1000及之后平台的陆续出现令人兴奋不已，但却始终未能形成规模经济、从而将成本缩减到理想范围（没错，最后一款Sparc执行效率更高，但价格却贵得离谱；尽管政府当局要求能耗过高的用户为碳排放过量状况付费，但即便如此最终的总体成本也远低于Sparc给数据中心带来的硬件支出）。

互联网经济的泡沫最终烟消云散，Sun公司决定将手中已经建成的大型设备集群转化为“商业化”计算硬件业务。总而言之，Sun在硬件业务方面押下了错误的赌注。

Sun所创造出的生态系统堪称伟大，他们只是未能建立起真正符合企业需求、能够激发用户购买欲望的产品。作为Sun成果的最终持有者，甲骨文充分燃尽了生态系统中的每一分潜力，蚕食或者毁灭掉与之相关的一切其它企业，从而创造出仅属于自己的高利润替代产品。

甲骨文在一份典型的简要公开声明中，承认某些业务及政治问题拖延了Java 7的发布进度。“众所周知，由于各种业务及政治问题的影响，最新版本的推出被迫延期。”

不过我们必须突破Sun的财务难题，继续将关注重点放在Java周边系统身上。Sun出尔反尔地公布了Java标准化计划，并创造出属于自己的“标准化”委员会，即Java社区进程组织。该组织最初的建立目的在于为实力雄厚的Java参与者们打造一个共商大事的平台，而且随着时间的推移其发展也逐渐步入正轨。然而如今Sun已经成为甲骨文毋庸置疑的附属，后者则直接忽略掉委员会的各种规则、粗暴行使着自己的一票否决权。

Java社区进程的发展为何受阻？问题不在于开放性，而在于利益争夺。尽管当时我是以旁观者的身份看热闹，但仍然清楚记得Sun在参与EJB3项目时遭遇的窘境。为什么Java发展进度会一落千丈？这是由于Sun与甲骨文双方需要将购买或者开发出的产品整合到应用程序服务器当中。一旦新的JavaEE规范出台，他们也必须保证自己能在市场上率先做出反应。

即使是在同一家公司内部，协调好单一产品的发布都绝非易事，更不用说在多家公司之间了。幸运的是，企业合并给事情带来了转机。我一直认为Java

社区进程并不是Java进度落后的主要原因。

将Saucer分离出来

如今，Sun只留存在我们的记忆中，而甲骨文则成为真正的老板。然而为什么Java新版本的发布仍然如此迟缓？最简单的解释是，Java项目规模过于庞大。大项目往往行动缓慢且充满风险。为了解决这个问题，让我们看看如何帮助Java“减肥”。

首先，甲骨文必须克服自身对客户端技术的过分依赖。当然，Swing与甲骨文还拿不出JavaFX的有力继承方案，毕竟在现代Web浏览器上开发出效果相同而又不引发新麻烦的机制绝非易事。不过甲骨文需要将客户牢牢束缚在自己的平台上——至少他们确实是这么做的。

目前我还不清楚JavaFX或者Java的客户端战略能给甲骨文带来哪些真正的优势。看起来甲骨文似乎设计出一种技术，用于同VB6、Flash或者某些4GL方案竞争。

在现代BYOD多平台环境下，每位与时俱进的高管人士都希望能在iPad上通过写写划划完成工作。为什么我们要用客户端来束缚服务器的能力？摆脱了客户端，甲骨文很可能不必再面对大量安全延误，并通过告别《Java零日安全漏洞》以及《如何在计算机中禁用Java》等头条为自己争取公关主动性。

不过事实上这种占用了大量资金投入的垂直技术平台应该被直接剔除并宣判其死刑，因为它对于解决主要矛盾根本毫无贡献。甲骨文甚至有可能将其添加到其它灾难性方案中，例如Java ME，并将其命名为Ordroid。如果甲骨文买下黑莓、将其重组为麾下的新部门并打出“未来的平台方案”以及“iPhone终结者”之类的唬人口号，投资者们也只会将其视为虚张声势的愚蠢方案。

简单来说，Java语言对于Java平台已经不再像过去那么重要。另一大负面因素在于，微软的介入削弱了Java的独有性，这种不利局面即使在2007年之后的实践协调活动中也未能得到扭转。对于甲骨文公司来说，将Java语言从Java平台中剥离出来并为其单独规划日程安排会更加轻松——毕竟甲骨文推出的开发工具既不属于Java相关业务中的主要组成部分，也没能得到Java开发人士的广泛支持。微软需要考虑Visual Studio版本是否能与下一代.Net以及C#版本相协调，但甲骨文则不必为此担心。

Java平台支持多种编程语言，从JavaScript到JRuby再到Scala不一而足。此外，以高性能及可扩展方式支持这类不同技术对于云计算非常重要。如果云计算将成为未来的发展方向，那么Java平台与甲骨文都需要提前做好准备。目前，Java与甲骨文已经对这一趋势表示默认。

在我们看来，对Ruby、Scala甚至Node.js的广泛支持已经成为Java平台的最大特色。然而正因为如此，目前Java平台更多被视为一种立足根基而非创新引擎。

在为Java平台选择合适的支持语言类型方面，我更信任Charles Nutter（JRuby/红帽）与Martin Odersky（Scala/Typesafe）而非Mark Reinhold（Java SE规范/甲骨文）。我对Reinhold先生绝无丝毫不敬之意，而且已经有证据表明众多协作尝试正在进展当中，不过等待Java语言或其它甲骨文附属项目的发展实在耗去了太多时间。

对于甲骨文在Java领域的领导权来说，这是充满挑战的一年。Sun当初做出的许多决定开始回过头给使用者带来困扰。我给出的答案是放弃Java客户端、将JVM与语言的发布周期分离开来，同时专注于将Java打造为一个平台而非万能性解决方案。■

JavaScript游戏开发实例指南

任何一种知名编程语言都可用于游戏开发（当然，某些语言会让游戏开发流程变得更加困难）。总而言之，游戏开发虽非易事，但也并不一定门槛过高。在今天的文章中，我们就一起探讨如何以简便的方式打造属于自己的游戏作品。

简介

对我来说，开发游戏不仅仅只是种爱好、更源于某种偏执。游戏开发往往是项复杂的工作，涉及的功能以及吸引玩家关注的因素越多、开发过程就越艰难。任何一种知名编程语言都可用于游戏开发（当然，某些语言会让游戏开发流程变得更加困难）。总而言之，游戏开发虽非易事，但也并不一定门槛过高。在今天的文章中，我们就一起探讨如何以简便的方式打造属于自己的游戏作品。

背景信息

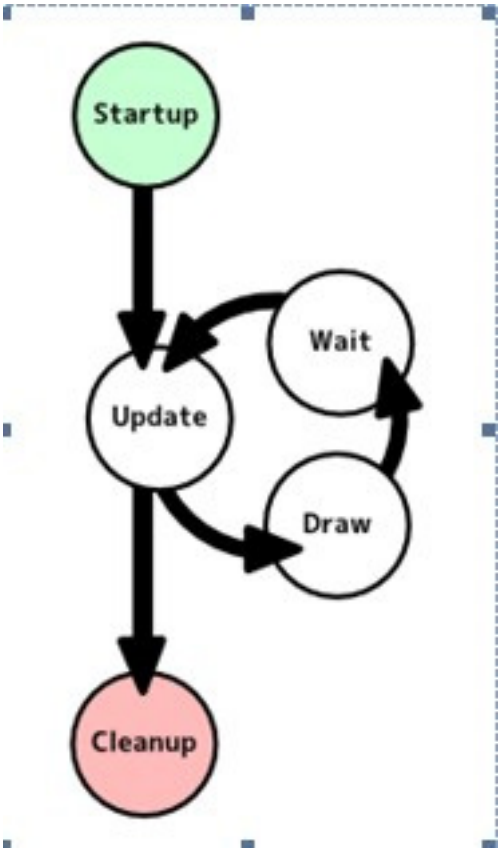
虽然没有明确证据，但JavaScript很可能已经成为全球使用率最高的通用语言。能取得这样的成绩，也要归功于互联网的迅猛发展以及HTML 5在平板及移动设备上得到的广泛支持。

有鉴于此，我决定在本文中向大家介绍如何利用HTML5与JavaScript进行游戏开发。

在开始之前，我们先来介绍关于游戏开发的背景知识：

Game Loop

游戏开发工作中最重要的元素被称为Game Loop，意为“游戏循环”，主要负责游戏的流程体系绘制工作。游戏循环的存在保证了游戏能够继续进行而不会被用户的操作所打断。



那么该如何建立游戏循环？在下面的示例中，我将按照大多数开发人员的直观感受首先创建后台进程，不过这其实并非最有效的解决方案。

```
while (!MyGame.stopped) {
    MyGame.update();
}
```

游戏循环的主要作用之一在于调用update函数，即负责游戏内容的更新(包括图形及对象位置等)。

如果我们使用While语句来调用update函数，那

么一方面浏览器很可能阻断该函数，另一方面我们也无法控制update函数的调用频率、导致执行效率低下。换言之，我们将无法控制游戏的FPS比率。

另一种常用的游戏循环创建方式则是利用setInterval函数，通过这种方式我们能够更轻松地控制update函数的调用频率：

```
MyGame.fps = 60;
MyGame.update = function () {
    // Move game parts };
// Start the game loop
MyGame._intervalId = setInterval(Game.update, 1000
  / Game.fps);
// To stop the game, use the following:
  clearInterval(MyGame._intervalId);
```

以上代码中出现的“FPS”指的可不是第一人称射击，而是每秒帧数，也就是我们常说的帧速率（也被称为帧频率）。它负责控制设备每秒钟显示不同画面内容的次数。人眼与大脑每秒钟最多只能分辨10到12幅独立画面，不过为了达到流畅的画面效果（例如游戏与电影），FPS值需要设置得更高一些。

第一部分–从EASEJS入手

尽管Game Loop的创建看似简单，但以此为基础进行游戏开发实在麻烦多多。下面我将利用EASEJS库所提供的Game Loop及其它相关功能打造自己的游戏作品。

EaselJS提供了一套非常直接的解决方案，能够容纳丰富的图形元素并与HTML 5 Canvas进行交互。它还提供一套专用API，旨在为Flash开发者提供更为熟悉的使用体验且同时能够识别JavaScript。它包含一套完整的分层显示列表、一套核心交互模型以及帮助类等等，从而显著简化了Canvas的使用流程。

Canvas元素是HTML 5中的组成部分，支持动态、脚本化2D图形与位图图像渲染功能。这是一套低级程序化模型，只能更新位图图形、不具备内置场景图像。

EASEJS下载地址 – http://www.createjs.com/#!/EaselJS

创建新的HTML页面并在代码头中添加以下内容。

```
<script type="text/javascript" src="js/easeljs/geom/
  Matrix2D.js"></script>
<script type="text/javascript" src="js/easeljs/geom/
  Rectangle.js"></script>
<script type="text/javascript" src="js/easeljs/
  events/EventDispatcher.js"></script>
<script type="text/javascript" src="js/easeljs/utils/
  UID.js"></script>
<script type="text/javascript" src="js/easeljs/utils/
  Ticker.js"></script>
<script type="text/javascript" src="js/easeljs/utils/
  SpriteSheetUtils.js"></script>
<script type="text/javascript" src="js/easeljs/
  display/DisplayObject.js"></script>
<script type="text/javascript" src="js/easeljs/
  display/Container.js"></script>
<script type="text/javascript" src="js/easeljs/
  display/Stage.js"></script>
<script type="text/javascript" src="js/easeljs/
  display/SpriteSheet.js"></script>
<script type="text/javascript" src="js/easeljs/
  display/BitmapAnimation.js"></script>
```

```
<script type="text/javascript" src="js/easeljs/display/
  Graphics.js"></script>
<script type="text/javascript" src="js/easeljs/display/
  Bitmap.js"></script>
<script type="text/javascript" src="js/easeljs/display/
  Text.js"></script> <span style="font-size: 9pt;">
  </span>
```

将Canvas元素添加到Body部分，这样Canvas就会成为游戏的容器环境。

```
<canvas id="gameCanvas" width="600"
  height="100"></canvas>
```

现在我们开始处理EASEJS，首先要做的是创建一个新的对象类型——Stage。

所谓stage，是指专为显示列表设置的基础Container。每当我们调用其Stage/tick方法，它都会将显示列表呈现在目标canvas当中。

```
stage = new createjs.Stage(id("gameCanvas"));
```

在完成了Stage对象的初始化之后，我们可以开始游戏中添加更多元素。接下来是创建Bitmap对象（也是EASEJS中的组成部分）用于接收图形路径，然后利用addChild函数向Stage中添加新的Bitmap对象。当所有游戏元素都被添加到Stage对象当中后，我们需要通过调用createjs.Ticker.setFPS(30)来启动游戏循环；最后登记“tick”事件（这样每当tick事件被触发时，游戏都会进行一次更新）。

Ticker会根据设定好的时间间隔提供集中式tick或心跳广播。监听器可订阅该tick事件以确保每隔固定时段即获得通知。

需要注意的是，tick事件的时间间隔被称为“目标间隔”，一旦CPU利用率较高、其间隔时间就会受到影响。Ticker类使用静态接口（例如Ticker.getPaused()），且不应被实例化。

```
var stage, image, fpstext;
function init() {
 stage = new createjs.Stage(id("gameCanvas"));
image = new createjs.Bitmap("assets/hill.png");
stage.addChild(image);
createjs.Ticker.setFPS(30);
createjs.Ticker.addEventListener("tick", tick); }
```

现在我们需要执行tick函数。在本文的示例中，我们将把图形向右侧移动10个像素，一旦图形达到600像素位移距离，我们则将其返回到初始位置（x=0）。在改变了图形的属性之后，我们需要调用stage.update()函数来刷新canvas。

```
function tick() {
image.x += 10;
if (image.x > 600)
image.x = 0;
stage.update();
}
```

在观察结果之前，我们再向stage中添加一个新元素，即text元素。我希望在Stage中直接显示FPS数值。要在stage中增加文本信息，我们要向Init函数中添加以下

```
fpstext.x = 0;
fpstext.y = 20;
stage.addChild(fpstext);
```

而要添加当前当前FPS速率，则需要向以下代表添加至tick函数当中：

```
fpstext.text = Math.round(createjs.Ticker.
  getMeasuredFPS()) + " fps";
```

点击此处查看演示效果。

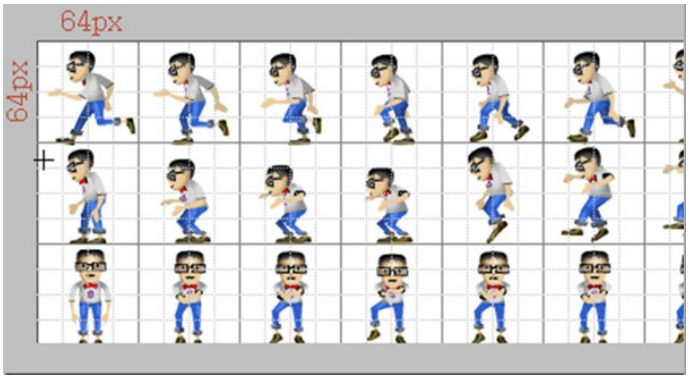
第二部分–Sprite

在第一部分中，我们了解了关于EASEJS的基本知识，包括向Stage中添加元素以及如何移动这些元素。在第二部分中，我们将一同利用Sprite以多种方式显示我们的角色对象（例如跳、跑以及其它小动作）。

Sprite专指没有经过照明、效果等转换通道处理而是直接绘制得到的2D位图显示目标。Sprite通常被用于显示游戏内的简单信息，例如体力条、生命数量或者当前分数等文本内容。

一幅Sprite图形中包含若干子图像，我们可以通过指定图形坐标的方式显示其中的特定子图像。

首先，我们先将Player Sprite载入到游戏当中。需要强调的是，请务必确定图像已经载入完成后再启动游戏循环。在范例中，我登记了onload事件，只有载入事件结束后、我才能执行start函数。



首先，我们先将Player Sprite载入到游戏当中。需要强调的是，请务必确定图像已经载入完成后再启动游戏循环。在范例中，我登记了onload事件，只有载入事件结束后、我才能执行start函数。

```
var stage, player, playerImage, _action;
var playerWH = 64;
var frequency = 4;
function init() {
    playerImage = new Image();
    playerImage.src = "assets/Player.png";
    playerImage.onload = start;
}
```

在第一部分中，我们利用Bitmap对象在stage中显示出图像。但既然现在开始使用Sprite，我们就需要引入有能力处理Sprite图形的不同对象才行。

SpriteSheet – 旨在将所有与sprite列表相关的属性与方法都打包保存。一套sprite列表中包含一系列图像（通常是单独的动画帧）结合而成的大型图像。举例来说，如果一小段动画中包含有八幅100x100的图片，那么整个sprite列表的尺寸则为400x200（每行四幅图片、共两行）。

在创建了Stage对象之后，我们还要创建playerSprite作为SpriteSheet对象的代表。所有传递至SpriteSheet构造函数的数据都用于定义三类关键信息：

单张或多张图片的使用方式。

单独图形帧的显示位置。此数据可通过两种方式表示：作为连续规则网格中的等尺寸帧一同处理，或者作为不规则（非顺序）网格中可变尺寸帧被单独定义。

同样，动画也可通过两种方式表现：作为一系列连续动画帧（拥有开始与结束帧定义）中的■

本文未完，请点击下面连接阅读全文：
http://developer.51cto.com/art/201308/405867.htm

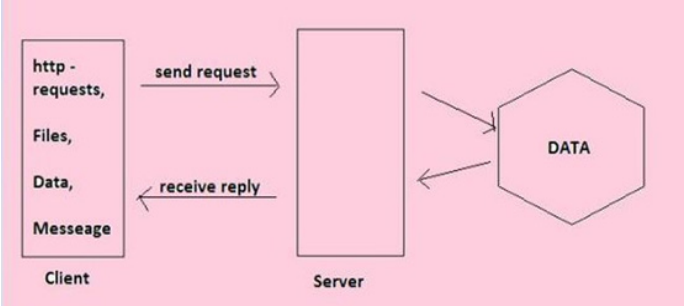
HTML5 WebSockets node.js实例教程

■ HTML 5 中WebSockets是极其重要的部分,它的一个好处之一是减少了不必要的网络流量。

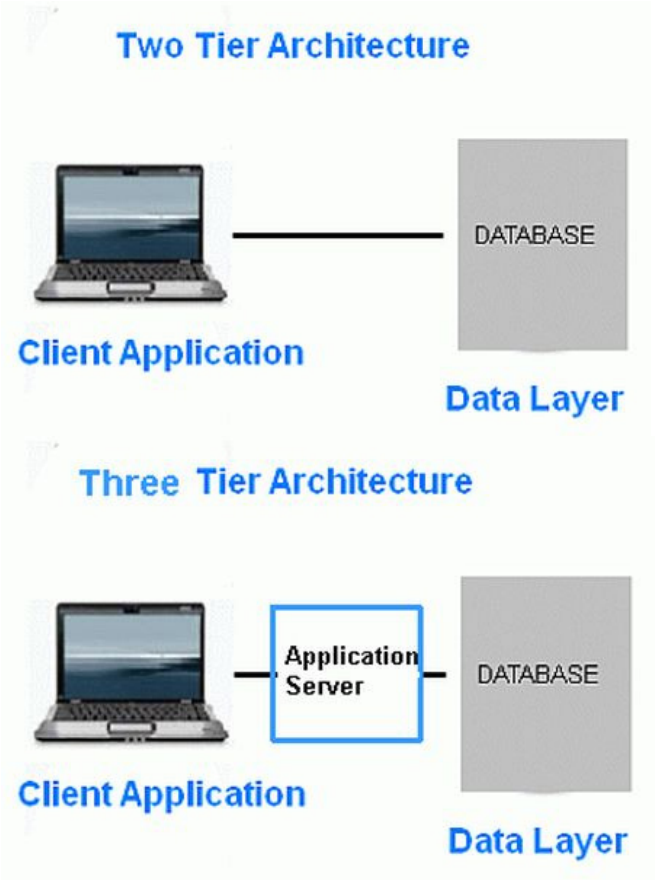
HTML 5 中WebSockets是极其重要的部分,它的一个好处之一是减少了不必要的网络流量。它主要是用于在客户机和服务器之间建立单一的双向连接。这意味着客户只需要发送一个请求到服务端，那么服务端则会进行处理，处理好后则将其返回给客户端，客户端则可以在等待这个时间继续去做其他工作，整个过程是异步的。

WebSockets应用的场景是很广泛的，当需要实时显示数据的时候（比如股票，实时聊天，对战游戏等），用户最期望的是不需要去刷新页面，否则获得的用户体验是不佳的。而在最新的HTML 5标准中，出现了两个解决该方面问题的规范，一个是服务端推送事件（Server Sent Events），另外一个是WebSockets，其中服务端推送事件由于在IE 中不能获得原生支持，因此不在本文讨论范围之中。本文着重讲解HTML 5中WebSockets的相关知识。首先我们还是来复习下在HTTP 的网络世界中的基本架构，这是下一步学习的基础。

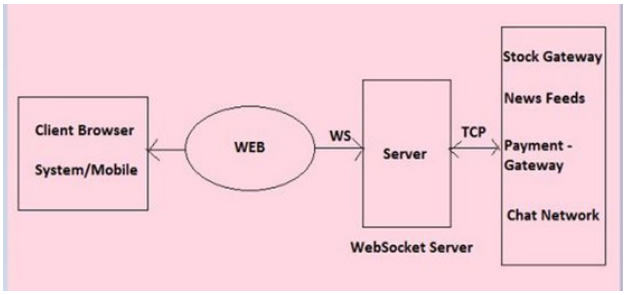
客户－服务器模式是网络世界中最经典的模式了，能允许多个用户通过前端图形界面的方式去，如下图：



通常，客户服务端模式分为两层和三层。在三层结构中，通常是出现在web应用中，中间一层使用了各类的中间件服务器，分别如下两图所示：



我们来看下HTML5 WebSocket的架构图,如下：

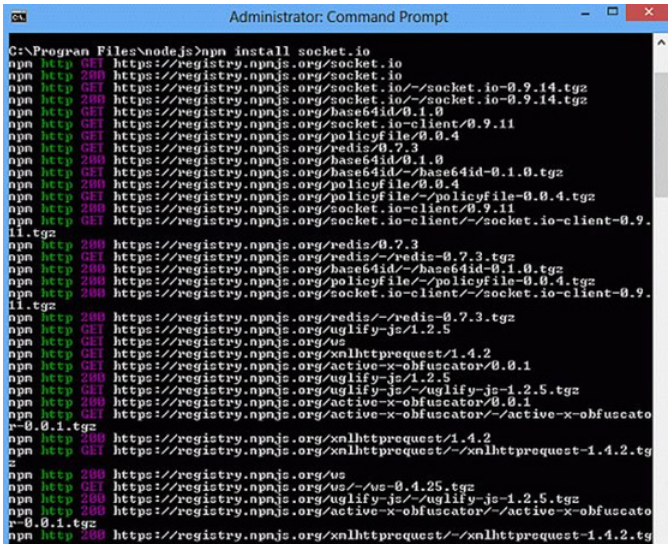


在HTML 5 WebSockets标准中，仅仅定义了一些列的API,这些API用来在客户端和服务端之间建立“socket”连接。下面我们来学习使用node.js来进行WebSockets的学习。

首先，到node.js的官网下载node.js，我们选用windows 版本就可以了。



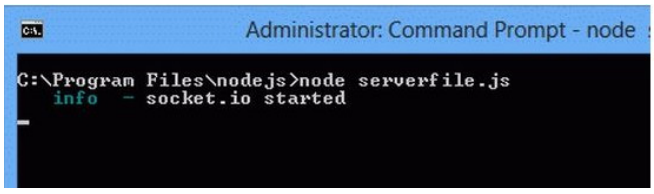
下载安装包后并选择socket.io，这只需要在 安装好的nodejs目录下，运行命令install就可以了，如图：



然后，我们在nodejs目录下建立一个简单的js文件，命名为serverfile.js，然后编写代码如下：

```
var http = require('http');
var io = require('socket.io');
var yourserver = http.createServer(function
  (request, response) {
response.writeHead(250, { 'Content-Type': 'text/
  html' });
response.end('Your WebSocket server is running');
  }).listen(22222);
var yoursocket = io.listen(yourserver).set('log', 1);
yoursocket.on('connection', function (client) {
  client.on('message', function (data) {
console.log('Client Message: ', data);
var current = new Date().getTime();
client.emit('YourMessageResponse', data + '->' +
  current);
client.on('disconnect', function () {
console.log('Your Client disconnected');
});
 }); });
```

然后重新在nodejs的目录中输入如下图的命令，以启动服务端：



这样，服务端就已经启动了，下面来编写客户端，客户端其实就是一个简单的html页面，代码如下：

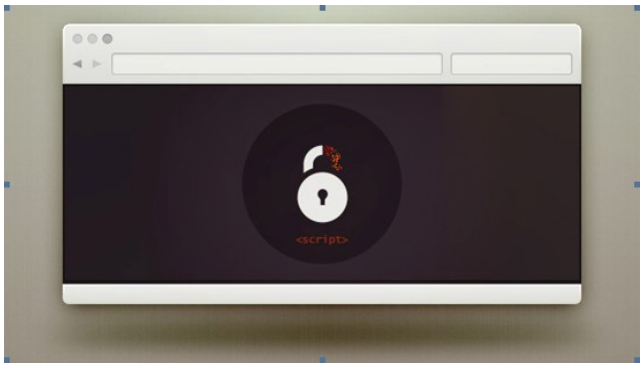
■

本文未完，请点击下面连接阅读全文：
http://developer.51cto.com/art/201308/407192.htm

在浏览器中攻破JavaScript到底有多轻

□ 核子可乐译

如何挫败攻击者的入侵企图。



Jesus Rodriguez发问：

我的问题涉及JavaScript安全性。

请大家想象一下，如果您正在某种认证系统中使用JavaScript框架（例如Backbone或者AngularJS），而且需要端点验证机制以保障安全。这样的情景似乎没什么难度，毕竟决定权始终掌握在服务器手中、而且在我们进行任何操作前都会通过验证检查其是否符合规定。

但是如果整个流程不涉及服务器，对安全性的要求还能不能成为现实？

举例来说，如果您拥有一套客户端路由系统而且打算通过保护具体线路来巩固用户登录的安全性。在这种情况下，大家需要首先向服务器发送ping指令来查询自己是否有权访问受保护的线路，接着再执行后续操作。问题在于，当我们ping向服务器时，系统会将服务器响应结果保存为一个变量。这样在下一次使用这条专有线路时，系统会自动检查您是否已经登录（而不再重复ping向服务器），而仅仅通过上一次响应结果决定您的申请是否通过。

听起来似乎有机可乘，那么通过修改变量来达成访问到底可不可行？

坦白的讲，我对于安全及JavaScript方面的知识了解不太多，但这样变量只存在于模块模式中的专有部分而非全局环境当中。这里只涉及获取方而无关于设定方，如果看来，我们似乎有机会针对其加以破解。

发送机密数据

Joachim Sauer的回答（获得66票支持）：

破解过程非常简单：任何依赖于客户端的安全机制都会按我们的要求进行操作，因此只要攻击者获得控制权、接下来的入侵任务将手到擒来。

大家可以在客户端上进行安全检查，但检查的对象实际上只涉及“缓存”内容（这是为了避免与服务器之间的高昂往返成本，如果客户端已经了解到响应内容为‘否’，那么直接从缓存内读取结果即可、无需再与服务器进行通信）。

如果大家希望保障一组用户的安全，请确保这些用户的客户端永远无法获取服务器响应信息。因为一旦大家向客户端发送“机密数据”，那么即使是加上了“请不要显示其内容”的指示，攻击者也能够轻松通过禁用对应代码对请求内容进行检查。

如大家所见，这项答案并没有真正提到任何与JavaScript及浏览器有关的细节。这是因为无论您所使用的是哪一种客户端，其基本概念都是相通的。事实上，无论是胖客户端（即传统客户端-服务器应用）、传统的Web应用程序还是承载着大量客户端JavaScript信息的单页面应用，都适用于前面所提到的假设。

一旦数据离开服务器端，大家必须做好攻击者有能力对其进行全面访问的准备。

另辟蹊径

Benjamin Gruenbaum的回答(获得17票支持)：

在首先阅读Joachim的答案，然后再转到这里。前一条答案涵盖了客户端安全漏洞的一般性出现原因，现在我们一起来看Benjamin对于解决这项问题的建议。

这是一项无需手动将每条请求发送至服务器端进行验证的客户端-服务器通信安全规划：

大家仍然允许服务器拥有掌控权，服务器也仍然对客户端发来的请求进行验证，但整个过程都以透明化形式进行。

假设通过HTTPS协议防止MITM（即中间人）攻击。

客户端与服务器间进行首次握手时，服务器会针对客户端生成一个公共密钥，同时生成一个私有密钥，从而构成一套非对称式加密方案。客户端在本地保存的是服务器的“公共”密钥，并利用一条不会保存在任何位置的安全密码进行加密。

客户端现在已经离线。客户端希望执行受信操作。客户端通过输入密码获取服务器的公共密钥。

客户端现在会根据对数据的了解执行操作，并在服务器对应公共密钥的辅助下对每项操作都进行了加密。

当客户端处于在线状态时，会向受公共密钥加密的服务器发送自身ID以及所有相关操作。

服务器负责对操作进行加密，如果加密格式无误，则信任这些操作确实来自客户端。

备注：

我们绝不能将客户端的密码保存在任何位置，否则攻击者将有机会取得密钥并实施自己的恶意操作。这套方案的安全性依赖于服务器为客户端所生成的密钥的完整性。当需要使用该密钥时，服务器仍然会对客户端进行验证。

我们实际上仍然需要仰仗服务器的安全机制而非客户端，客户端执行的每项操作都需要经过服务器的验证。

我们可以在Web Workers中运行外部脚本。请注意，每一条JSONP请求如今都有可能引发严重的安全问题。我们需要不计一切代价保护密钥。一旦密钥丢失，攻击者可以轻而易举地伪装成用户并实施恶意活动。

这样的处理方式符合“不ping向服务器”的要求。攻击者无法简单利用伪造数据模仿HTTP请求，只要密钥没有泄露、一切恶意活动等都不可能实现。

Joachim的回答仍然正确，我们事实上仍然将所有验证工作留给了服务器。惟一的区别在于，现在我们不必在每次操作时都要求服务器验证密码。大家只需要在提交内容或者更新数据时才直接面向服务器。我们所做的一切工作都是为了在客户端保存

受信密钥，并且保证客户端重新验证的顺利进行。

对于单页面应用（例如Angular）而言，这是一套相当通行的控制方案。

我之所以将服务器的公共密钥称为“公共”，是因为RSA等方案使用过同样的形容字眼，而且意味着该密钥属于敏感信息、应该受到严格保护。

我不会把密码保存在任何存储机制当中。我需要确保用户每一次运行离线代码时都必须提交“离线”密码。

不要使用自己的加密机制：请务必使用知名度较高的可靠加密库，例如斯坦福大学的认证机制。

请认真阅读并掌握这套推荐方案。在大家将此认证机制推向实际操作之前，请先向安全专家咨询其可行性及潜在影响。这项问题非常严重，实施过程并不轻松而且很容易在执行中出现错误。

最关键的一点在于，千万不要让其它脚本介入到页面当中，这意味着我们只能通过Web Workers引入外部脚本。我们绝不能轻信其它外部脚本，因为其中很可能潜伏着可能在用户登陆时截获密码的恶意内容。

如果大家无法完全肯定脚本安全性或者推迟其执行（也就是说，该脚本不应该成为事件的访问对象，而只作为同步代码），请务必向用户发出提示而且不要使用内嵌密码字段。另外，不要把密码内容保存在变量当中——再次强调，只有百分之百确信用户的计算机不存在安全漏洞后才能这么做（当然，这又会涉及到服务器对客户端的验证）。

我还是要再次提醒各位，我们要以怀疑的态度审视客户端，这一点在Joachim的回答中也已经表达得非常清楚。在新机制下，我们只不过可以在工作开始之前摆脱ping服务器这一步骤。

任何人都不可信

nvoigt的回答（获得7票支持）：

游戏行业有一句名言：“客户端掌握在敌人手中”。任何运行在外部安全区域（例如服务器）中的代码都有遭到侵袭的可能。要不要真正运行我们提供的“安全代码”完全由客户端决定，用户只能在有限的范围内做出选择。不过在本地代码方面，我们至少可以通过自动混淆机制保护代码内容；也就是说，必须是拥有出色编程能力的攻击者才有可能跨过这道障碍，从混淆机制的重重迷雾下看清JS与纯文本信息的真面目。

攻击活动所需要的准备工作非常简明：一台代理服务器外加一个文本编辑器，基本上已经足够了。诚然，攻击者需要对编程拥有相当程度的了解，但利用文本编辑器修改脚本内容可比直接编写一套可以执行的注入代码简单得多。■

开发框架工具推荐

10个新的很有用的开发框架

10款最新且超实用的Web开发框架

10款无需编码的网站开发利器

分享18个常用的网站性能测试工具很棒的HTML5开发框架和开发工具

响应式设计的6个免费测试工具

只会编程的程序员没有前途

□ Ali R.Tariq

比尔-盖茨曾说：“学习编程锻炼你的思维，帮你更好地思考，它能形成一种思考方式，让你更好地思考一切，这对所有领域都有帮助。”

于是，编程可以扩大你的思维，编程可以强化你的能力，编辑是可以赚大钱的……

于是，编程被过份强调了。

让我澄清一点：编程，被过份强调了。

有一种言论很流行，那就是软件代码是未来的语言，更进一步说，编程是未来的必备技能。作为个体的你，如果想在未来有立足之地，最好学点编程。

这种言论的后果是：

1、进入学校的人，或者准备开始寻找工作的人，他们会认真考虑进入软件开发行业，哪怕他们真正的兴趣在其它地方。

2、已经毕业的、没有编程背景的人觉得自己低人一等，相比于拥有编程背景的同事来说，觉得自己容易受到排斥。

依我之见，这两点都是对趋势的过激反应，看起来世界经济变得越来越以代码为中心。问题在于：它相信经济、世界和未来会以代码为中心。

真是这样吗？不是的。

只要我们处在前Singularity时代（Singularity是微软提出的概念操作系统），只要人类统治我们的经济、社会、政治决策，我们的全球生态系统就会一直由人类需求驱动。其它都是工具或者语言，它用来强化个人。

当中包括代码。

我们应该将编程看成促使你成功的技能，它不是诅咒，也不是万能药。现在是黑客通才时代，任何人都可以学点编程技能，但拥有其它职业。这些人并非为编程而编程，它是工具，可以提供价值。在我教导的MBA学员中，有一个团队曾制作非常漂亮的幻灯片。我问他们谁设计的，他们指着团队的一个成员，这是一个心脏外科医生。他告诉我，在成为医生之前曾当过图形设计师。图形设计师转型心脏外科医生，真不可思议!

如果他学点编程呢？如果知道用HTML5和CSS3制作网站呢？如果会调整HPP代码呢？如果他能下载和使用jQuery插件呢？我想，他的能力会为病房、病人带来巨大增值价值。

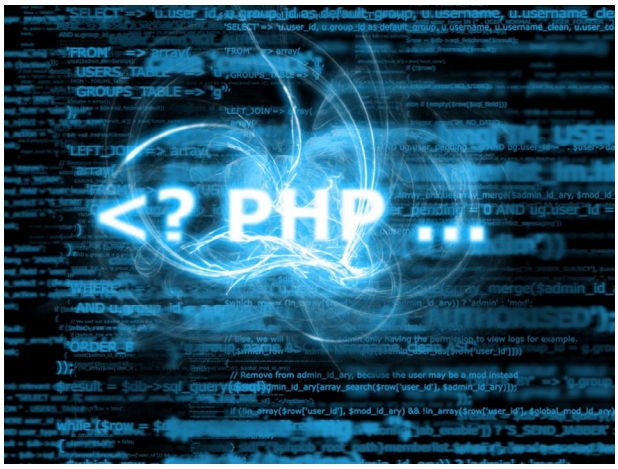
21世纪，我们应该理解到生活的许多方面是复合互联的，技术是彼此融合的。一旦我们理解到这点，会会知道作为个体、团队、组织的我们应该如何去解决那些纠缠人类的问题。单是成为Java EEE专家、或者Django巨头、或者程序语言奇才，这已经不够了。要真正推动世界前进，脱颖而出，成为明天的问题解决者，变得真正有创造力，就要拥有一些新东西：深度和宽度。总的来说，就是要擅长一些其它的学问。

它告诉我们：

1、如果你是一个编程人员，学点其它领域的技能和知识。这会增加可能性，让你将编程技能用在其它方面。

2、如果你不是一个编程者，但有其它领域的技能，你可以学点基本的编程知识。这会增强你的能力，让你站在新层次上解决问题。它还可以通过新的方式利用你的天赋。

作者Ali R.Tariq，MyFamilyTales联合创始人■



最近几年主要参与开发PHP面向对象的Web项目。PHP语言目前的框架主要是采用与JAVA类似的MVC开发模式，含有

权国清：谈PHP发展现状

权国清，河南科技大学本科毕业生， 5年多PHP软件项目开发经验。现就职于郑州金擎电子科技有限公司，带领团队，参与开发郑州本地O2O电子商务（有卡优惠、九鼎团购、九鼎商城）等商业项目以及各类在线支付（支付宝、手机支付、银联在线支付、）及TTG（U联生活）、微信接口、退款功能等。目前承担WAP站和网站新版（优惠券或有卡优惠）等项目的程序设计。PHP5.5在前几个月发布了、Google开始支持PHP。种种动作，再次预示了PHP黄金时代的来临！近日，我们电子邮件联系到他，请他谈一谈PHP发展的现状和前景。以下是交流信件原文：

你最开始做程序觉得最难的是什么？

高中的时候开始更多的接触计算机，大学的时候自然选择了网络技术专业，对这方面的东西比较感兴趣。所以毕业以后就坚持打算做程序员，去做软件开发方面工作。毕业后就在西安软件园进行了一段时间的实习，实习结束时期，遇到08年全球金融危机，没有找理想的工作，只是从事和计算机相关的工作。后来有一个机会，就到无锡（国家）软件园实训，再一次的技术进修，对相关技术做了更系统的学习，这之后，才正式开始做开发，到无锡六月智能软件有限公司参与PRPC欧美银行软件开发。我觉得那段时间应该是最难的，不是很累，但是心情上总是在摇摆，定不下来，跟现在做项目遇到问题思考完全是两回事。

您是怎么开始开始关注PHP的？开发的第一个PHP项目是什么？您最满意意的PHP项目是什么？

最早的时候，其实是在一家公司做网页设计师，公司那时候刚好缺PHP程序员，自己之前也是做Java开发的，就转到PHP，开始做了PHPer，也是从那个时候开始深入php。第一个PHP项目是《全球品牌搜索网》，不过这个项目不是最先上线的，后来参与的一个“磐谷汽配轮毂网” 比我刚才说的那个项目提前上线。最满意的是现在的项目“有卡优惠”，这个项目主要是和TTG进行合作的刷银联卡的进行优惠的项目。毕竟自己的在技术方面做得越来越成熟，想法也比几年前的完善多了。

PHP、Python和Ruby这三种开发语言几乎是同时出现的，都是十分有特点、优秀的开源语言。这个月TIOBE编程语言社区公布的语言排行帮上，PHP势头迅猛，上升到了第五，很明显PHP的发展势头比其他两种语言更好，您认为这和PHP在网站开发的立足点与PHP语言本身的特点有什么关系？还是有什么其他的因素影响？

以上3种语言都是相当优秀的脚本语言，现在越来越多的新公司或者新项目使用PHP，这使得PHP相关社区越来越活跃，而这又反过来影响到很多项

目或公司的选择，形成一个良性的循环。就我目前了解到的情况，PHP是国内大部分web项目的首选，而且有很多公司从其它语言（如ASP，JAVA）转到了PHP。合适的就是最好的，PHP的快速，开发成本低，周期短，后期维护费用低，开源产品丰富（这点可以影响到很多项目的决策），这些都是另外两种语言无法比拟的。另外，在很多大项目中，似乎正在走向与PHP不同的发展方向，让我们拭目以待。

您对于PHP开发工具的现状满意么？您目前工作的团队使用的是什么开发工具？

个人认为，一个优秀的框架能在很大程度上提高工作效率，尤其是对大中型项目。当然如果对项目有足够的把控能力，框架也并非是必须的，毕竟PHP的框架跟ROR的情况有所不同，多而混乱，而学习一个框架是需要时间的。像你的问题中提到的，很多老的PHP程序员对框架不屑一顾，认为自己对项目有足够的把控能力，不需要框架这么累赘的东西，然而从长远来看，一个优秀的框架可以延长一个项目的寿命，降低维护和二次开发的成本。在一个项目中，没有一个优秀的框架是很难想象的。目前主流的PHP框架有很多的，但是没有哪个框架能完全适合一个大型的项目。目前的开发团队正在使用的是ZEND框架，极大地提高了团队的工作效率。

有关PHP网站漏洞的消息是很多的。您认为这其中的原因和PHP语言本身的特点有关么（相对于如PHP网站数量、PHP代码编写习惯等其他因素）？PHP的安全有改善的空间么？

刚开始做开发的时候，自己其实是不怎么喜欢使用开源框架，感觉到自己写的类库文件可以到处调用，挺方便的使用，其实是因为自己那时候太低级了。后来开始接触zend、tp、yii等框架，通过学习简单的使用，发现可以节省时间、灵活性较强。对程序员来说可以有清晰明确的分类归类，减少代码量。目前网站安全一直是一个烦琐的问题，它主要是服务器安全、网络安全以及代码的质量，我不想说PHP这门语言不安全。目前PHP有很多优秀的开源产品，例如ZEND、TP、YII等，即使在开源的情况下，这些产品也保证了足够的安全性。PHP的门槛很低，初学者一两月的时间就可以入门，而这些“快速入门”的程序员，容易忽略代码质量安全问题。安全问题的改善，极大部分取决于越来越多的PHP程序员对这个问题的重视程度，从而加强代码的质量。

最后，谈谈您对PHP未来发展的看法。

不论对什么样的语言，面向对象更多的是一种思考问题的方法，没有必要为了面向对象而去面向对象。PHP最初的面向对象，可以简单地看作是函数的升级，曾经被很多其它语言的开发者取笑，但PHP5的面向对象已经足以满足大部分WEB项目了。一门语言要想生存下去，就得去适应技术发展的潮流，并不断做出调整，在这方面，PHP的团队充满了活力，相信未来的发展空间是相当大的。

最近几年主要参与开发PHP面向对象的Web项目。PHP语言目前的框架主要是采用与JAVA类似的MVC开发模式，含有丰富的开发库，支持多数据库驱动。未来PHP开发框架同时需要具备几个特点：

采用PHP的新特性，例如命名空间、匿名函数；

拥有更丰富可扩展的开发包；

文档丰富；

社区活跃。每个PHP开发人员在选择的时候都应该根据自己情况选择最合适自己的开发框架，从而方便快速地进行项目开发，提高工作效率，节省时间、减少成本核算。 ■

系统级编程语言性能大PK D语言成首选

■ “C/C++已经统治系统编程很久，除了ObjectiveC之外语言都无法获得很高的关注。有人用多种系统级语言编写了同样的地图生成工具来测试他们的性能，包括D(DMD,LDC,GDC)、Go(GCC-Go,6g)、Haskell(GHC)和Rust。

"C/C++已经统治系统编程很久，除了ObjectiveC之外语言都无法获得很高的关注。有人用多种系统级语言编写了同样的地图生成工具来测试他们的性能，包括D(DMD,LDC,GDC)、Go(GCC-Go,6g)、Haskell(GHC)和Rust。

相比C/C++，这些语言都原生支持了诸如垃圾回收这些高级特性，也因此无一能达到C/C++的运行速度。这其中表现最差的是原生Go语言编译器6g，只有Clang22%的速度，而表现最好的是基于LLVM的D语言编译器LDC，达到了79%。

由于原生就使用了LLVM编译，Rust成为各语言原生编译器里最快的一个，但也只达到了45%。从结果来看，D语言一定是首选。由于D语言许多特性都依赖垃圾回收，如果需要关闭垃圾回收而又要保持良好的使用体验，则推荐Rust。"（摘自：Solidot）

Go，Rust，Haskell和D四语言做关卡基准测试

作者在为一款游戏制作随机关卡生成器，尽管这个游戏是用C++编写，且是模块化，但是类似的关卡生成器却可以用更高级的语言来编写。因为C++并不是最好玩和最丰富的语言，所以作者打算选择其他语言替代。

“如使用特别依赖迭代和条件句的简单Roguelike关卡基准，然后粗略模仿他们的真正生成器。”点击此处获取代码。作者认为Haskell语言的是最易读的版本：

```
roomHitRoom Room {rPos=(x,y), rw=w, rh=h} Room
  {rPos=(x2, y2), rw=w2, rh=h2}
 | (x2 + w2 +1 ) < x || x2 > (x+w+1 ) = False
 | (y2 + h2 +1 ) < y  || y2 > (y+h+1 ) = False
 | otherwise = True
```

检查新生成的room与原有的room是否有冲突，如果有冲突就放弃（这是强攻关卡技巧；我们真正的引擎要更为复杂，但仍然遵循同样的原则）。大多数剩余的时间花在了随机数字生成上，所以给基准加点料，使其像常速那样具备单独语言的随机数字生成器（例如，这些基准与作者的目标相关，而与你的无关。）。

PS：所有从Haskell分离出的部署现在都使用XorShift PRNG方法，比较起来更合理。

事实上，只有LDC D的使用XorShift，因为DMD的用XorShift运行起来比用C rand()要慢一些。一个部署XorShift PRND的Haskell版本可能是最受欢迎的。

结果如下（C语言版本用于对比）。

编译器 LDC | Clang | GCC | Rust | GDC | DMD | GCCGo | GHC | 6g

时间： 0.412| 0.280 | 0.390 | 0.620 | 0.680 | 0.770 | 0.850 | 1.05 | 0.544

最快(%)： 68% | 100% | 72% | 45% | 41% | 37% | 33% | 27% | 51%

LDC，Rust和Clang所使用的LLVM版本是3.2.GCC，GCCGo使用的是GCC 4.7.3版本，GDC使用的是 GCC 4.6.4版本。Rust 是0.8-pre (9da42dc 2013-07-17 04:16:42 -0700) 版本，GHC是 7.6.2版本，DMD是2.036版本，6g(Go)是1.1.1版本。它们运行的都是03，-opt-level=3可用于Rust，-release可用于DMD，-funbox-strict-fields可用于Haskell。**D现在运行的时候同样有-inline和-noboundscheck，所以其速度从1.36s增加到了1.59s。

D是目前测试过的最快的非C语言。在这里不得不表扬LDC编译器的设计师们。作者非常期待着一年以内再次运行这些基准，看看Clang，LLVM D和LLVM Rust运行得怎样，并对它们做出比较。

Rust，虽然最初比较慢，但是通过XorShift PRNG，其速度已经得到大幅提升。但是为了使Rust的句法适应语境，所以Rust的灵活性要受一点点影响；要通过堆形分配的向量，不过需要引用myFunc(&mut myVector)，接收到它的函数要在其类型签名有myAlias:&mut~[myVector] ，和fn myFunc(myAlias:&mut ~[myVector]) {..} 一样。与C比较则是

```
void myFunc(struct MyType myAlias[arrayLength])
{
..
 }
```

Rust版本看上去有点拜占庭的味道。还没有用过Rust的人可以看看，这里给出7种指示符： @ (收集垃圾的堆形分配), ~ (独特的堆形分配), & (借给堆形/堆栈分配的指示符), * (原始的C指示符, 仅限于不安全代码), 以及前三种符号的变体(事实上，不确定是否真的有&指示符，所以可能总共只有6个)。注意，具有可变值的指示符和没有可变值的指示符是截然不同的。

作者还用堆栈分配型向量对Rust版本做了基准测试，但是速度方面的差异不明显。不过发现Rust中的堆栈分配型向量目前有点冗长，因为它不允许未初始化的值，所以不得不创建一个对象实例，作者想要一个所有值都被设置为零的数列，然后用上述创新来填充向量。希望，不久的将来，Rust也能学习Go，将所有数值自动设置为零，或者至少提供这样一个可选项。目前，它看上去就像这样：

（显然，这种想法已经成为可能（https://news.ycombinator.com/item?id=6094819）。希望下次升级的时候，Rust教程里能将其记录下来。）

```
et emptyr = Room {X:0,Y:0,W:0,H:0,N:0};
let emptyt = Tile{X:0,Y:0,T:0};
let emptyl = Lev{TS : [emptyt,..2500] , RS :
  [emptyr,..100] };
let mut ls : [Lev, ..100] = [emptyl,..100];
```

这里面有很多不必要的代码，如果是使用堆栈分配型数列较多的大型项目，这就非常累赘。

Go给人的印象很深刻，虽然不如D，但是作为一种相对比较新的语言，已经非常不错了。默认的PRNG在速度方面有缺陷，所以可改用XorShift。分号的自由使用是一种很好的体验，就作者个人经验而言，它很适合写命令式代码。

尽管作者花了很长时间做优化，但是他仍然喜欢Haskell的性能，特别是它必须得携带一个包■

本文未完，请点击下面连接阅读全文：
http://developer.51cto.com/art/201307/405621.htm

用什么编程语言的程序员赚钱最多

■ 曾尝试用不同方法截取数据，得出针对每个编程语言而不是组织的人口统计概况。有关使用不同编程语言的开发人员的刻板印象比比皆是，我很好奇它们如何与现实挂钩。

几个星期前，我在博客中描述我如何使用Git的元数据和RapLeaf的API来构建GitHub组织的人口统计概况（查看博客文章点此，查看每个组织的数据点此）。

我也曾尝试用不同方法截取数据，得出针对每个编程语言而不是组织的人口统计概况。有关使用不同编程语言的开发人员的刻板印象比比皆是，我很好奇它们如何与现实挂钩。

从每个编程语言使用人群的基本信息如年龄、收入、性别开始分析，根据我手上既有的数据，要做到这一点并不困难：

我采用GitHub对每个数据资源库编程语言组成情况的估算。例如，GitHub估计某个项目使用了75%的Java语言；

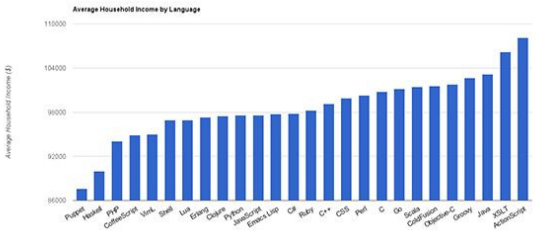
我从中得知每个项目中使用比例超过50%的是什么编程语言，某个项目中使用这一主要语言的开发人员是谁，并把他们的收入累加；

我进而筛选出得到大于100收入数据点的编程语言。

以下便是收入情况的统计，根据家庭平均收入由低到高排列：

Language	Average Household Income ($)	Data Points
Puppet	87,589.29	112
Haskell	89,973.82	191
PHP	94,031.19	978
CoffeeScript	94,890.80	435
VimL	94,967.11	532
Shell	96,930.54	979
Lua	96,930.69	101
Erlang	97,306.55	168
Clojure	97,500.00	269
Python	97,578.87	2314
JavaScript	97,598.75	3443
Emacs Lisp	97,774.65	355
C#	97,823.31	665
Ruby	98,238.74	3242
C++	99,147.93	845
CSS	99,881.40	527
Perl	100,295.45	990
C	100,766.51	2120
Go	101,158.01	231
Scala	101,460.91	243
ColdFusion	101,536.70	109
Objective-C	101,801.60	562
Groovy	102,650.86	116
Java	103,179.39	1402
XSLT	106,199.19	123
ActionScript	108,119.47	113

同样的数据以图表的形式体现如下：



多数的排行结果大致符合我的预期：

-Haskell是种非常学术的语言，因此在收入方面并不可观；

-PHP是种较为容易掌握的语言，方便非专业的或初级编程人员使用，也因此收入相对不高

-Java和ActionScript则被认为是高级的语言，多用于企业软件的开发，因此收入颇丰。

-另一方面，我对处在低端和高端的某些语言并不十分了解，比如XSLT, Puppet, and CoffeeScript , 也不清楚它们排名前后的原因。

我们也要看到使用这些数据得出结论的局限性:

-这些项目是开源的，无法适用于封闭程式开发人员的报酬；

-Rapleaf数据不涉及总收入信息，样本可能因此有所偏差；

-我忽略了年龄、性别等因素对造成数据分布偏态的可能性；

-我没有分析所有的GitHub数据资源库，作为样本的用户资料可能不具代表性。

综上，即使在绝对值上有所偏差，这仍然是比较不同编程语言间相对收入差别的一个开始。■

从零到整：.Net编程结构知识汇总

专题入口

专题内容列表

- .NET 数据类型概述
- .NET数据类型汇总
- .Net运算符和基础语句
- .Net方法概述和分类
- .Net索引器和迭代器
- .NET 面向对象基础
- .Net异常处理

我拒绝接受的几个最佳编程实践方法

□ 外刊IT评论/文

import类，而不是import整个包

在很多语言里，这通常是一种被推荐的做法，有些甚至是必须的。如果是在C++里，这还算是有点意义，因为更少 #include 意味着更快的编译速度，然而，这种意义仅体现在需要花很长时间去编译的大型项目中。

而对很多像Java这样的语言，这毫无意义。因为它不影响编译的时间，所有你得到的回报只是花更多的努力来维护你的import语句。虽然IDE可以帮助你做这些事情，但你仍然需要时不时的多点几次鼠标/键盘，在版本控制系统里多留几条变更记录，干扰你的代码审查。有啥实际用处？向官僚机构表明代码很规范，无它用途。

面向接口编程

这项编程法则要求程序员定义接口，并针对接口来编程，而不是针对实现编程。理由非常简单：容易开发第二种实现，易于测试(真的吗？)，更有效的使用代码。

问题就出在你不能凡事都按照这个原则。我个人认为，如果一个方法需要有多个实现，那使用接口是不二选择。但除此之外，如果你仍遵守的话，除了增加代码量，增添麻烦外，不会有任何好处。而且，把一个类重构成接口和它的实现并不困难，事实上是非常简单，所以，为什么一开始就要写接口呢，当需要时把它改造成接口也不晚。

禁用某种语言功能

在很多企业、组织使用的编码规范中，你会发现各种各样的类似于“不要使用goto语句”，“不要使用三元操作符”等规则。

如果一种语言的某种语法并未标志为“deprecated”，为什么不让人使用？当然，要正确的使用！即使像 goto 这样的语法同样可以让代码更可读、更易理解——只要你能以正确的方式、用在正确的地方。

使用Setters/getters，禁止public属性

这是最著名的Java风格，Java里任何公有属性都是不提倡的，任何属性都应该通过 setters 和 getters 操作，不允许有任何质疑。有些共用框架更加强化了这些。每次当我看到一个5年前的老类里只有一些私有属性和公有的无聊的 setters 和 getters ，我都会奇怪这是要干嘛？是为了增加代码量？是为了预防将来有可能出现意外的属性值修改？但如果真的有人修改了，能起到何预防效果？

单个返回语句

有人说多个返回语句会让代码变复杂。我发现却正好完全相反。当方法/函数在退出之前需要做一些收尾工作时，单一return语句会让函数更简单，但在其它很多情况下，这反而会让事情变得复杂，你需要添加额外的if-else来处理各种非正常退出情况。

尽量责任分离

我这里主要是针对“尽量”。有些人把这做到了极限，甚至有些变态。没错，把大的复杂的问题拆分成小的简单问题，这很好。但拆的太小就会引起新的问题。如果你把一棵树砍成牙签那么大小的块，你得到的就是一堆垃圾。

有些问题本身很复杂，你无法通过拆解来让它简单。

为了让这篇文章有个比较积极的结尾，下面是我认为的放之四海皆准的最佳实践方法：

-做任何事情都要有个理由

-如果你做的未能符合预期，重做，替换方法或修正

-扔掉垃圾通常是你最应该做的事情-不论这垃圾造价多高■

我也谈谈.NET程序员工资低

□ subwayline13/文

我从2011年下半年预谋转型，2012春季正式转型到iOS，看了《经过本人 6 年.net 工作经验证明 .net 工资确实比 Java 低》这篇文章，一下子有很多感慨。

我不好意思算我干了多久了C#了，反正从VS2002开始接触，从VS2003干起，来帝都混饭吃的时候用的VS2005，VS2010的时候离开.NET平台，基本上都是小公司。回过头来看看，有一种浪费青春的感觉，因为干这些年C#，个人着实没有什么进步，走了很多弯路。当然，我不否认我资质平庸，但是不能否认的是.NET这个技术真是入门容易，提高却很难。

有些无关紧要小事你会发现印象特别深刻，下面开始回忆往事和无节操吐槽。

我学C#的时候买一本叫《C#本质论》的书，讲的应该是C#1.0吧，为什么我要提这本书呢？因为这本书讲了很多语言细节，一次我在公司，我给大家讲obj as ClassType 和(ClassType)obj的区别的时候大家投来了真诚和赞许的目光，让我得意了好一阵子。现在再看，这TMD不是基础嘛。看了这本书的另一个好处是，一般笔试C#语言上问题，比如多态，重载的问题之类基本没问题了，有时候面试官还觉得我语言掌握的不错。这本书貌似不是什么名著，后来也没在系统的看过C#语言的书了。但是，我接触到的很多C#程序员基础差的令人发指，比如有一次一个ASP.NET页面，测试的时候没问题，但是上线的时候页面经常有诡异的错误，最后排查原来是把属性全设置成static了，还有SQL语言都是直接拼接的，导致后来被SQL注入。

顺便吐槽一下webform，用C#之前，我是写PHP的，所以分的清前端后端，但是一些直接用webform的人来说，他们是分不清前端后端的，只会拖控件。一次一个朋友让我去个小公司帮忙，看看能不能解决他们的问题。到了之后才知道，原来他们用嵌入了activex，activex上的某些操作会导致页面刷新，所以treeview 控件一下子就失去状态了，后来我试图给他讲POSTBACK，解释为什么失去了状态，但是他一直纠结于为什么嵌入了这个控件treeview就不行了呢，以前是可以的，后来提议让我给他完成这个功能，看他们迷惑的眼神，我只想赶快离开。

下面吐槽一下自己，很长一段时间我的SQL还是很差的，以前光研究什么NHibernate, linq to sql, entity framework。后来也是一次面试，考数据库内容的时候被鄙视了之后才看SQL SERVER 2008 INSIDE系列，先从那本SQL语言基础看起，看了几章就真心体会到自己以前的数据库水平有多烂。

还有一段时间，很迷恋JAVA技术，设计模式

啊，框架啊，还都是工业标准啊，牛逼的要死。现在呢，websphere就是落后笨重的代名词，这都是弯路。

所以在回到为什么.NET程序员工资低，其实最简单了，因为水平不够，为什么干了六年还是没到月薪一万以上，因为确实没到那个水平，比如我吧，我把大部分时间浪费在了学习各种框架的API上了，这有什么用呢？如果把六年的时间去研究数据库，去研究前端技术，那么在帝都，15K以上绝对不成问题。

其实纠结于.NET和JAVA程序员那个工资高的意义不大，因为JAVA烂程序员也不少，他们也拿着不高的工资。而前面提到的文章的作者也不必纠结，厚积薄发，跳槽之后薪水翻翻的例子更是不计其数，在帝都，.NET程序员的薪水还是给的起的，比如elong就是.NET的，据说都是15K以上都比较普通，每年至少13.8个月工资在加额外奖金，现在通胀了，不知道他们涨工资了没有。JAVA也好，.NET也好，都不过是工具，不过一个是oracle牌，一个是微软牌。决定木匠薪水的是手艺而不是用的什么牌子的锤子，换个牌子的锤子就不会干活了？即使微软牌锤子倒闭了，你就当不成木匠了？

但是，.NET相比JAVA，确实门槛低，门槛低的结果就是混进来很多低水平程序员。大量低水平程序员在一起工作，如果想提高的话真的很难，要走很多弯路。这么多年来，微软的技术都是只有微软才用的好，就很说明问题。而微软的技术为了降低门槛，经常设计出非常复杂的框架，比如我前面吐槽的asp.net webform，入门很简单，但是想掌握的话，真要费一番功夫，但是等你掌握的时候，其实发现没什么用，一般情况下，用不着自己实现一个webcontrol，而且设计一个好用通用的控件，还是很难的，还不如直接用PHP简单灵活。后来推出MVC，算是浪子回头了。

所以我劝.NET程序员不要把眼光局限于.NET平台，而且不要人云亦云，一定自己尝试，比如很多牛人都喜欢黑C++，我以前人云亦云，跟着黑C++，但是为了转iOS，学了C++，发现根本不是那么回事，C++那么设计是有他的道理的，是权衡利弊的结果。我现在看C# vs JAVA的言论都觉得很没意思，的确我认为C#在语言上比JAVA好一点，但是这点优势根本无法撼动JAVA的地位。还有就是少研究框架，少追新技术，新技术的失败率还是很高的，多研究一下本质的东西，做B/S的比如前端技术，HTTP，数据库，说来惭愧，很长时间我都没明白均衡负载是怎么玩的。■

Web相关热门下载

20款绝佳jQuery倒计时脚本和插件

30个HTML5和CSS3表单设计

40个让网站更加友好的jQuery插件

推荐15个适合用于Web设计的字体

27个jQuery工具提示Tooltip插件

10款无需编码的网站开发利器

开发者必备的6款源码搜索引擎

10个超有用的网页设计工具和资源

■ 编者按

要在企业IT领域顺利晋升绝非易事，不过让自己在竞争当中脱颖而出则更是难上加难。通过本篇文章，我将为大家带来职业规划建议、希望能为IT专业人士的发展之路带来启示。

保障IT生涯顺利晋升下一级别的14种方式

要在企业IT领域顺利晋升绝非易事，不过让自己在竞争当中脱颖而出则更是难上加难。通过本篇文章，我将为大家带来职业规划建议、希望能为IT专业人士的发展之路带来启示。



今年对于技术领域的专业人士们可谓风调雨顺——六月份技术市场上的新岗位增加了10%，而且这一趋势将延续至整个2013年。这条新闻对于渴望在 IT阶梯上更进一步、变换工作岗位或者寻找新工作的技术人员们来说无异于一针强心剂。不过为了顺利实现这些愿望，大家还需要进一步提升自己在雇主眼中的价 值。

根据CIO、职业顾问以及技术专家们的观点，单单做好手中的工作还远远不够。大家需要尽量展示自身在专业技能成长与水准提升方面的能力与潜质。切实可行的发展路线图取决于各企业的实际情况，但只要遵循本文的指导、相信大家的晋升之路会走得更为顺畅。



与主管及人力资源负责人积极交流

让你的老板与人力资源负责人体会到你对成功的渴望及奋不顾身的精神。根据企业专业发展顾问David Brookmire的建议，如果您身处一家中型企业，人力资源部门应该拥有明确的技能水平划分，例如IT专家级、IT主管级等等，直接与人力资源负责人进 行交流及询问，或者从公司的门户网站中获取有用的信息，”Brookmire指出。这是推动我们职业发展规划的理想起点。

寻找理想的导师

导师在大多数情况下能为我们带来助益，甚至成为攀登企业职级阶梯的成功关键。他们能够根据棘手

的职场状况为我们提供建议，或者透露企业内部不为人知 的选拔机制。他们既拥有丰富的知识，又愿意与我们在技术及业务方面分享心得。总之，如果您现在还没有得到前辈的扶持，请马上查阅公司相关规定并为自己找上 一位。

“辅导关系至关重要，”Energy Plus公司CIO Hugh Scott表示，“学习与成长的最佳方式就是在工作中与前辈建立起正式或者非正式的师生关系。”



个人成长与发展能力就在我们身边的生活当中，而非仅仅局限在课堂之上，”MD ANdrson癌症治疗中心副总裁兼副CIO Keith Perry指出。作为一位程序员/开发者，大家能够找到很多自己感兴趣同时又有助于锻炼IT专业技能的项目。

大家不妨将自己的技能投入到非盈利性技术项目当中，这样的参与完全是出于热情而非回报。本地UseNet开发者组织也相当常见，其中的每位成员都为自己所使用的自主技术感到骄傲与自豪。不过请当心，志愿活动有种神奇的传播魔力。

深入了解公司业务

“我为IT专业人士提出的关键性建议在于从自己从事的行业中找到工作激情，并深入了解与业务相关的一切知识——而非只局限于IT本身，”MD Anderson癌症治疗中心副总裁兼副CIO Keith Perry表示。

了解业务的运作机制能帮助大家更好地将个人技术能力运用在创新及解决业务难题当中。“虽然IT部门的本职工作在于为业务策略提供支持，但支持也同样 能够为企业指明可行的发展方向，”Authentic Leadership Global公司CEO兼创始人Tana Heminsley解释道。



培养自己的人脉体系

大多数工作岗位都以相关技术为核心延伸人脉体系，为了在职场上更具灵活性，大家需要跳出自己的小圈子。无论是企业活动、开发者见面会还是社交媒体平 台，都可以成为我们结交值得信赖的专业人士的理想场合。“大家必须学会主动，积极接触结交那些制定决策的高层人士并与之建立伙伴关系。技术人员往往日常紧 张，并因此忽略了培养交际圈的重要性，这会给未来的个人发展造成不利影响，”David Brookmire表示。

“人脉体系的作用可谓不可或缺。虽然与人结交也属于一种可以通过学习来掌握的技能.....■

本文未完，请阅读原文：
http://developer.51cto.com/art/201308/407528.htm

编者按

我职业生涯起始的公司是神州数码网络公司（DCN）。如果将google或者twitter定义为优秀公司，那么DCN显然处在平庸公司的行列。好在DCN继承了老联想的底子.....

程序员：浮躁世界，我思考

1.（一）

又是一个闷热的周六，灰霾就像粘在天空一样，依旧笼罩着这个城市。带着小宝，我们三人游荡在蓝色港湾的儿童城中，为小家伙的未来打算着。老婆和丈母 娘穿梭于一个又一个玩具店，而我，因为推着童车，不便走来走去，就带着小宝静静地欣赏楼下冰场里玩耍的孩子们。很快，我们的注意力就被冰场上两个正在进行 对抗训练的小男孩吸引住了。小宝快乐地随着冰球转动脖子，双眼紧紧地捕捉着黑色的冰球，不时发出愉悦的『喝彩』；而我，默默地看着两位『男子汉』一次次跌 倒，又一次次爬起，对抗虽然异常激烈，但任何一方都没有脏动作，这足以让真正的冰球赛场上的成年人汗颜。

连日来，对环法自行车赛兴奋剂丑闻的追踪报道，让曾经的英雄们纷纷落马。成人世界里，人们对于名利的渴望，对成功的追逐已经远远超出了能力或是道德 的底线。我们不得不承认，这个世界正变得越来越浮躁。会爬的恨不得立即会走，踉踉跄跄走着的，又恨不得能撒腿就跑。前两天我在卓越亚马逊上随便查了查『小 时学会』和『天学会』，结果发现分别有65条和157条结果之多。『Java程序设计24小时轻松掌握 』，『21天学会javascript』，『30天学会Visual C++』，...看着那列出的一大堆需要花很大力气学习和使用的语言，我不得不感慨这种浮躁已经无处不在。

小宝还在咯咯地乐。那种发自内心的，如雪花般洁净的快乐让我由衷地羡慕。如果不是上周日LP跟我严肃地讨论了一下在未来我如何更好地履行父亲的责任，此刻（周六的下午）我应该端坐在办公室里思考某段代码该如何去写，而不是在这样一个放松的环境下，任由我的思想翱翔。

我们讨论的结果是，一周里大礼拜我能不去公司工作就不去，并且工作日最好有两天赶在七点前回到家，这样，可以大大增加我在家中陪伴小宝的时间，提升 她对『父亲』这个角色的认知。在北京这样一个拥挤到几乎处于崩溃边缘的城市，以公司到我家长达三十五公里的距离来算，晚上七点前回到家就意味着至少提前两 个从公司出发，还得祈求一路上 1）没有事故（小概率事件），2）没有管制，3）没有幽灵堵车。两个小时的车程对很多北京人来说还在忍受的极限内，但对我来说，开车上/下班的时间超过一 小时我就已经浑身难受，更遑论两小时这一让人绝望的数字。所以，我决定在提前回家的那两天搭乘地铁，这样，只消花一个半小时，我就能回到家中；而且，在地 铁里，我可以暂时远离代码和互联网，翻几十页和技术毫不相干的书。巧的是，上周不开车的两天上下班时间里，我翻的书是《长日留痕》——一本我买了很久，翻 了几次却都没看进去的小说。史蒂文斯先生，书中的主人公，利用一次旅行的机会回顾了他漫长的管家生涯。在整部书中，他始终在回忆过去的人生，并且

不断地探 讨一个严肃的问题：一位杰出的男管家究竟是什么样子的?

感谢这本书，让我开始严肃地回顾我自己的职业生涯。这思考断断续续游走了好几天，很多时候干脆藏身在潜意识中，以至于每每我工作累了，对着窗外中关村东路和成府路交叉的路口发呆时，它就会蹦出来，让我暂时从繁重的工作中解脱出来，在回忆中尽情徜徉。

2.（二）

教练一声哨响，把我暂时拉回。此时此刻，两个小家伙开始了一攻一守的对抗，场面也变得好玩起来——两个小愣头青从场地的两端以几乎相同的路线滑向中 圈，砰的一声，重重地撞到了一起，一同倒地。教练上前，把攻方的小家伙拉到一旁耳语几句，并比划了几下，随即恢复了比赛。两人从场地的两头快步冲向中圈， 守方将重心控制地很低，高速滑动中，目不转睛地盯着冰球。就在两人要接触的一刹那，攻方一个急停，然后迅速切向斜前方。由于扑得太猛，守方来不及调整，无 奈让攻方轻松绕过自己，打进一球。不知是看懂了比赛，还是单纯开心，小宝挥舞着双手，发出欢畅的尖叫声。

这就是教练的作用。他们告诉你什么是基本功，如何应对局势，以及一名优秀的球员应该具备什么素养。在一个人职业生涯的早期，一位好的导师比什么都重要。一晃我的程序员生涯已经过去了十年，在这十年里，我一直在不断探索一位优秀的程序员究竟该具备什么素养，如果当时有个教练能给予我指导，给予我探索的方向，那该多好啊。

可惜没有。软件行业没有教练，也没有拜师学艺的传统。也许曾经有，但自从程序员从手工艺人变成了工程师，教练或者师傅就不再重要。你基本不需要像手 工艺人那样具备从无到有做出点什么的技术，你需要的仅仅是解决某个或者某些特定问题的能力 —— 某种意义上说，这和流水线上的工人并无二异。这也导致了那些速成的图书的畅销，因为它们试图让你学习到的是如何完成任务，而非如何编程。当然，互联网正在粉碎这一切，程序员似乎又在回归手工艺人的传统，这很好。

虽然没有教练，但每个软件公司基本都会为新员工指定一名资深的同事提供『象征』意义上的“mentor”。我说『象征意义』，并不是否定mentor的作用，而是说这些mentor并非职业化的mentor，『传道授业解惑』的为师之 道，mentor们勉强在解惑上提供了些许帮助。当然，对于公司来说，mentor最重要的作用是成为一个榜样（Role Model），通过他们的工作态度，工作习惯，工作能力，对新员工，尤其是处在职业生涯早期的员工产生潜移默化的影响。

我职业生涯起始的公司是神州数码网络公司（DCN）。如果将google或者twitter定义为『优秀』公司，那么，DCN显然处在平庸公司的行 列。好在DCN继承了老联想的底子，还有一批有理想的做系统的牛人。Z君就是这样一个人。碰巧我被分在了他所在的团队。当我把『一位优秀的程序员究竟该具备什么素养』这样的问题抛给我的mentor，即将离职的X君时，他让我多看，多学Z君。之后，有意无意地，我都会多去劳烦Z君，他总会抛开手头的事，爽 快地为我答疑。为了解决某个问题，他经常工作到深夜；跟他讨论代码如果没揪到根上他绝不善罢甘休。拿code review来说，当你看到一堆逻辑上无比正确但写作上WTF的代码时，即使想骂娘，但屈服于release的压力或人情世故，你总会妥协。■

本文未完，请阅读原文：
http://developer.51cto.com/art/201308/405816.htm

Java程序员在写SQL程序时候常犯的10个错误

■ SQL是说明性语言而非面向对象或是命令式编程语言。在SQL中要写个查询语句是很简单的。但在Java里类似的语句却不容易，因为程序员不仅要反复考虑编程范式.....

Java程序员编程时需要混合面向对象思维和一般命令式编程的方法，能否完美的将两者结合起来完全得依靠编程人员的水准：

技能(任何人都能容易学会命令式编程)

模式(有些人用“模式-模式”,举个例子,模式可以应用到任何地方，而且都可以归为某一类模式)

心境(首先，要写个好的面向对象程序是比命令式程序难的多，你得花费一些功夫)

但当Java程序员写SQL语句时，一切都不一样了。SQL是说明性语言而非面向对象或是命令式编程语言。在SQL中要写个查询语句是很简单的。但在Java里类似的语句却不容易，因为程序员不仅要反复考虑编程范式，而且也要考虑算法的问题。

下面是Java程序员在写SQL时常犯的错误(没有特定的顺序)：

1.忘掉NULL

Java程序员写SQL时对NULL的误解可能是最大的错误。也许是因为(并非唯一理由)NULL也称作UNKNOWN。如果被称作UNKNOWN，这还好理解些。另一个原因是，当你从数据库拿东西或是绑定变量时，JDBC将SQL NULL 和Java中的null对应了起来。这样导致了NULL = NULL(SQL)和null=null(Java)的误解。

对于NULL最大的误解是当NULL被用作行值表达式完整性约束条件时。

另一个误解出现在对于NULL 在 NOT IN anti-joins的应用中。

解决方法：

好好的训练你自己。当你写SQL时要不停得想到NULL的用法：

这个NULL完整性约束条件是正确的?

NULL是否影响到结果?

2.在Java内存中处理数据

很少有Java开发者能将SQL理解的很好.偶尔使用的JOIN,还有古怪的UNION,好吧.但是对于窗口函数呢?还有对集合进行分组呢?许多的Java开发者将SQL数据加载到内存中,将这些数据转换成某些相近的集合类型,然后再那些集合上面使用边界循环控制结构(至少在Java8的集合升级以前)执行令人生厌的数学运算.

但是一些SQL数据库支持先进的(而且是SQL 标准支持的!)OLAP特性,这一特性表现更好而且写起来也更加方便.一个(并不怎么标准的)例子就是Oracle超棒的MODEL分句.只让数据库来做处理然后只把结果带到Java内存中吧.因为毕竟所有非常聪明的家伙已经对这些昂贵的产品进行了优化.因此实际上,通过将OLAP移到数据库,你将获得一下两项好处:

便利性.这比在Java中编写正确的SQL可能更加的容易.

性能表现.数据库应该比你的算法处理起来更加快.而且更加重要的是,你不必再去传递数百万条记录了。

完善的方法:

每次你使用Java实现一个以数据为中心的算法时,问问自己:有没有一种方法可以让数据库代替为我做这种麻烦事.

3. 使用UNION代替UNION ALL

太可耻了，和UNION相比UNION ALL还需要额外的关键字。如果SQL标准已经规定了支持，那么可能会更好点。

UNION(允许重复)

UNION DISTINCT (去除了重复)

移除重复行不仅很少需要(有时甚至是错的)，而且对于带很多行的大数据集合会相当慢，因为两个子select需要排序，而且每个元组也需要和它的子序列元组比较。

注意即使SQL标准规定了INTERSECT ALL和EXCEPT ALL，很少数据库会实现这些没用的集合操作符。

处理方法：

每次你写UNION语句时，考虑实际上是否需要UNION ALL语句。

4.通过JDBC分页技术给大量的结果进行分页操作

大部分的数据库都会支持一些分页命令实现分页效果，譬如LIMIT..OFFSET,TOP..START AT,OFFSET..FETCH语句等。即使没有支持这些语句的数据库，仍有可能对ROWNUM(甲骨文)或者是ROW NUMBER() OVER()过滤(DB2,SQL Server2008等)，这些比在内存中实现分页更快速。在处理大量数据中，效果尤其明显。

纠正：

仅仅使用这些语句，那么一个工具(例如JOOQ)就可以模拟这些语句的操作。

5.在java内存中加入数据

从SQL的初期开始，当在SQL中使用JOIN语句时，一些开发者仍旧有不安的感觉。这是源自对加入JOIN后会变慢的固有恐惧。假如基于成本的优化选择去实现嵌套循环，在创建一张连接表源前，可能加载所有的表在数据库内存中，这可能是真的。但是这事发生的概率太低了。通过合适的预测，约束和索引，合并连接和哈希连接的操作都是相当的快。这完全是是关于正确元数据(在这里我不能够引用Tom Kyte的太多)。而且，可能仍然有不少的Java开发人员加载两张表通过分开查询到一个映射中，并且在某种程度上把他们加到了内存当中。

纠正：

假如你在各个步骤中有从各种表的查询操作，好好想想是否可以表达你的查询操作在单条语句中。■

本文未完，请阅读原文：

http://developer.51cto.com/art/201308/406064.htm

12 岁的孩子教你使用 Python 开发游戏

Garfielt /译

这是一篇由教程团队成员Julian Meyer发表的文章,一个12岁的python开发人员。你可以在Google+和Twitter上找到他。

你可曾想过如何创建视频游戏吗？它不像你想象的那么复杂!

在本教程中，您将创建一个简单的游戏叫做兔子和獾，在这里作为英雄的兔子要保卫城堡反击进攻的獾群。

写这个游戏，你将使用Python。不，我指的并不是一条大蛇！:

Python是一种计算机编程语言。我们就本教程选择Python，Python是一种易于上手、有趣而且很容易学的语言。

如果你刚接触Python，开始之前先检出这本书 – 深入Python：怎样像计算机科学家一样思考，这将使你提升速度。

然后回到这里，准备开始—— 有一场小兔子和獾之间的战争即将发生。继续阅读将加入战争吧!

开始: 安装Python

如果你想在Windows电脑上尝试本教程，你需要安装Python，记住要安装2.7.3版本而不是3.3.0！运行完安装程序后，在开始菜单程序文件夹中会有IDLE，运行IDL。

如果你使用Mac，Python已经安装了！只需打开一个终端(/Applications/Utilities/Terminal.app)，输入python然后按回车。

注意：如果你从python.org安装Python(如果你想获得PyGame需要这样做)，那么你也需要在Mac上进入IDLE。它应该在“Applications/Python 2.7”文件夹内。

如果操作正确的话，你将会看到类似输出：

```
Python 2.7.3 (v2.7.3:70274d53c1dd, Apr  9 2012,
  20:52:43)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for
  more information.
>>>
```

注意：如果你想退出Python提示窗（三个尖括号提示:>>>），你可以通过在提示窗里输入exit()或按Ctrl+D。

一旦你在Python提示窗了，测试Python是否正常工作的话输入print 1+1，然后按回车或换行，它将输出2。你已经写了你的第一个Python程序。

现在你知道Python工作正常，为使用Python写游戏你需要安装PyGame。



PyGame是一个使写游戏变简单的Python库，它提供如图像处理、声音回放等你可以很容易添加都游戏中的功能。

到这来下载你系统适合的PyGame，确认你下载的是一个Python2.7版本。

注意：从上面链接下载的PyGame安装文件不能与Apple的Mac上默认安装的Python一起工作，为使用PyGame你需要从python.org上下载安装Python，或者你可以通过MacPorts安装Python和PyGame。

要检验PyGame安装是否正确的话，打开IDLE或通过终端运行Python，在Python提示符处输入import pygame，如果回车后没有输出的话你安装成功了。

如果，另一方面，输出了下附类似错误，PyGame没有正确安装。

```
Python 2.7.2 (default, Jun 20 2012, 16:23:33)
[GCC 4.2.1 Compatible Apple Clang 4.0 (tags/
Apple/clang-418.0.60)] on darwin
Type "help", "copyright", "credits" or "license" for
more information.
>>> import pygame
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named pygame
>>>
```

如果你遇到类似的错误，在论坛上发出来我将帮助你让它正常。

从文件运行Python代码

虽然你可以在Python提示符下运行短的Python代码，但如果你工作在一个大的程序(比如游戏)里，你可能想要将你的代码保存到一个文件，这样你就不必一遍又一遍地键入它了。

有几种方法可以运行Python程序文件。一种方法是使用一个纯文本编辑器，不如记事本(Windows)或TextEdit(Mac)。打开一个新的文 本文件，输入你的Python代码(像print 1 + 1)。然后保存它为asXXX.py(XXX可以是任何描述性的文件名称)。

使用Windows的话，通过双击这个文件来运行它。在Mac上，打开终端，输入python，然后拖动你保存的文件到终端窗口并按Enter。

另一种方法是键入代码类到IDLE，也就是你在这个教程中需要做的方式。运行IDLE – 只需在终端输入IDLE。 然后选择文，然后选择File\New Window，会出现一个你可以输入代码的文本编辑器窗口。你可以通过File\Save保存你更改的代码，通过运行Run\Run Module(F5)运行代码。■

本文未完，请点击下面连接阅读全文：
http://developer.51cto.com/art/201308/406711.htm



我突发奇想。我想通过电子邮件的方式对那些我非常感兴趣和非常敬重的程序员问10个问题。准备这10个问题我只花了5分钟.....

对九个超级程序员的采访

这篇访谈源自2006年，最先发布在波兰程序员Jaroslaw “sztywny” Rzesz ó tko (AKA “Stiff”) 的博客上。但是这篇博文现在找不到了。非常感谢他能授权我重新发布这个博文。

在一个炎热无聊的下午，我突发奇想。我想通过电子邮件的方式对那些我非常感兴趣和非常敬重的程序员问10个问题。准备这10个问题我只花了5分钟，这些都是我个人想问他们的问题，所以，我基本上没想太多要问他们什么。最后两个问题和编程没有什么关系，我就是想问题这些人的一些兴趣爱好。另外，不是每一个人都想回答我的，这是我第一次做“访谈”，所以，我犯了一些错误，一些问题没有得到回答。不管怎么样，我得到了很多很有意思的内容，所以，这对我绝对是一次很有意义的经历。

并不是每一个人都回了我的邮件，也并不是每一个人都同意回答我的这些问题，也许在我发布这篇文章后我会得到那些回答，但是我已经迫不及待想把这些东西发布了，所以，我可能会更新这篇文章（更新：2006年3月8日，我收到了Bjarne Stroustrup的回信） —— Jaroslaw

介绍

Linus Torvalds – Linux kernel 作者。

Dave Thomas – “Pragmatic Programmer”（注：douban) 和 “Programming Ruby”(注：douban) 以及其它一些优秀书籍的作者。 你可以在 这里 读读他对编程的一些想法。

David Heinemeier Hansson – Rails Framework 作者– 一个目前最新最热的Web开发框架。他的blog在 这里.（陈皓注：他也是37signals的领导人之一）

Steve Yegge– 他可能并不那么知名，但是他给了很多有意思的回答。他有一个很火的关于编程的 blog，他也是游戏 ” Wyvern” 的作者。（陈皓注：他最火的是去年在google+上对google和amazon的吐槽，06年他应该在google了）

Peter Norvig –Research Director at Google, 知名的 Lisper，AI书的著名作家，个人主页。

Guido Van Rossum – Python 发明者。

Bjarne Stroustrup – C++发明者， 个人主页。

James Gosling – Java 发明者。

Tim Bray – XML 和 Atom 规格说明书作者之一 个人博客 。

Q 1: 你是怎么学编程的？是从学校里学的吗？或者你没有上过学:)？

Steve Yegge

在我17岁的时候，我在HP的计算器中用他们的RPN 栈语言自学编程的。在这之前，我尝试过学习编程一两次，但都没有学成。HP 28c 和 48g 的科学计算器是一个很牛的东西，而且还有不错的文档。我搞了一本3D图形的书，并很费力地把其中的Pascal语言转成RPN栈语言，并用48g写了一个 3D的线框图渲染图。运行的还不错，在我买了PC和Turbo Pascal之后，我开始认真地学习编程。在我进入大学计算机科学专业之前，我已经是一个不错的程序员了。

我在华盛顿大学拿到了计算机科学学位，这绝对是有价值的，所以，我建议所有的程序员都应该得到计算机科学专业的学位。

Linus Torvalds

我没有在学校里学过编程，我在主要是读我自己想读的书，或是就直接去编程（一开始在Commodore VIC-20 学编程， 然后是 Sinclair QL上编程)。

当然，我觉得上大学非常有用。我没有去一个工科大学，我上了赫尔辛基大学，这是一个比较偏理论的大学，所以，那里的教育并没有那么多的编程的东西（编程只是很少一部分），这里大多数的课程都倾向于教一些基础概念的东西，如：复杂性分析。看上去很无聊，甚至有点浪费时间，但是我还是觉得这些课有用， 我对大多数课都还比较enjoy。所以，我觉得我可能在这些方面是一个比较好的程序员。

David Heinemeier Hansson

我学编程是从用HTML做我的第一个网页开始的。那时，我想当我的网页能动态地显示一些内容，所以，我选择了ASP和PHP。在做完这个网页后，我知道了怎么去编程，于是我开始我的计算机科学和商业管理学位的学习。

Peter Norvig

我是从高中和大学课程中学编程的，但是我还是觉得我自己学得更多。

Dave Thomas

我是在高中学编程的。我完全地迷住了，我对编程爱得无法自拔，然后，我开始挑选那些提供软件开发课程的大学。最终，我去了伦敦大学的帝国学院。第二 年我就开始学习软件开发的课程了，那绝对是非凡的，学生和教员在一起工作把教材做得更好，每一个人都可以从中学到很多。这些课程给了我难以置信和非常雄厚 的软件开发背景。我在那里读到了博士，最后去创业了。

关于“我是怎么学编程的”这个问题，我的回答是“我现在还在学编程”。我认为好的程序员一生都在学编程。这并不是去学一门语言或是一个代码库，好的程序员会对他们的编程技艺一年又一年地精益求精。

Guido Van Rossum

我去的那个大学有一个大型主机和很多不同的计算机课程。这对我很重要。■

本文未完，请点击下面连接阅读全文：
http://developer.51cto.com/art/201308/406600.htm

数据即代码,我和小伙伴们都惊呆了！

■ 几个小伙伴在考虑下面这个各个语言都会遇到的问题：问题：设计一个命令行参数解析API……

几个小伙伴在考虑下面这个各个语言都会遇到的问题：

问题：设计一个命令行参数解析API

一个好的命令行参数解析库一般涉及到这几个常见的方面：

1) 支持方便地生成帮助信息

2) 支持子命令，比如：git包含了push，pull，commit等多种子命令

3) 支持单字符选项、多字符选项、标志选项、参数选项等多种选项和位置参数

4) 支持选项默认值，比如： port选项若未指定认为5037

5) 支持使用模式，比如：tar命令的-c和-x是互斥选项，属于不同的使用模式

经过一番考察，小伙伴们发现了这个几个有代表性的API设计：

1. getopt()：

getopt()是libc的标准函数，很多语言中都能找到它的移植版本。

```
//C
while ((c = getopt(argc, argv, "ac:d:")) != -1) {
  int this_option_optind = optind ? optind : 1;
  switch (c) {
  case 'a':
    printf ("option a");
    aopt = 1;
    break;
  case 'c':
    printf ("option c with value '%s'", optarg);
    copt = optarg;
    break;
  case 'd':
    printf ("option d with value '%s'", optarg);
    dopt = optarg;
    break;
  case '?':
    break;
  default:
    printf ("?? getopt returned character code 0%o ??", c);
  }
}
```

getopt()的核心是一个类似printf的格式字符串的命令行参数描述串，如上面的"ac:d:"定义了"a"，"c"，"d"3个命令行参数，其中，a是一个标志符不需要参数，"c"和"d"需要跟参

数。getopt()功能非常弱，只支持单个字符的标志选项和参 数选项。如果按上面的5点来比对，基本上只能说是勉强支持第3点，其他几项只能靠程序自己来实现了，所以，想直接基于getopt()实现一个像git这 样复杂的命令行参数是不可能的，只有自己来做很多的解析工作。小伙伴们看过getopt()之后一致的评价是:图样图森破。

2. Google gflags

接着，小伙伴们又发现了gflags这个Google出品C++命令行参数解析库。

```
/C++
DEFINE_bool(memory_pool, false, “If use memory pool”);
DEFINE_bool(daemon, true, “If started as daemon”);
DEFINE_string(module_id, “”, “Server module id”);
DEFINE_int32(http_port, 80, “HTTP listen port”);
DEFINE_int32(https_port, 443, “HTTPS listen port”);

int main(int argc, char** argv) {
    ::google::ParseCommandLineFlags(&amp;argc, &amp;argv, true);
    printf(“Server module id: %s”, FLAGS_module_id.c_str());

    if (FLAGS_daemon) {
      printf(“Run as daemon: %d”, FLAGS_daemon);
    }
    if (FLAGS_memory_pool) {
      printf(“Use memory pool: %d”, FLAGS_daemon);
    }
    Server server;
    return 0;
}
```

小伙伴们看了后不由得感叹“真心好用啊”！的确，gflags简单地通过几个宏就定义了命令行选项，基本上很好的支持了上面提到的1，3，4这几项，比起getopt()来强多了。对于类似cp这样的小命令，gflags应该是够用了，但要达到git这种级别就显得有些单薄了。

3. Ruby Commander

接下来小伙伴们又发现了Ruby Commander库：

```
//Ruby
# :name is optional, otherwise uses the basename of this executable
program :name, ‘Foo Bar’
program :version, ‘1.0.0’
program :description, ‘Stupid command that prints foo or bar.’ ■
```

本文未完，请阅读原文：
http://developer.51cto.com/art/201308/406687.htm



编者按

51CTO记者采访到了Moco框架创始人郑晔。关于Moco，和记者分享了更多的其诞生过程的设计理念、成长过程的思考、未来发展的期望。还对未来几年Java技术的发展情况和Java编程风格的现状和未

MOCO作者郑晔:谈MOCO框架技术和Java未来

MOCO是一款基于Java开发、用来简单搭建模拟服务器的程序库。MOCO的目的，旨在解决企业级系统集成测试等方面的问题。简化传统Jetty或Tomcat等应用服务器上部署一个新的JAR所需要的耗时工序，同时，MOCO的API设计采用了内部Domain Specific Language（DSL）的方式，提升了代码的可读性。与此同时，还将新的Java编程风格融入其中、并不断加入新的功能来满足更多企业项目需求。

为此，51CTO记者采访到了Moco框架创始人郑晔。关于Moco，和记者分享了更多的其诞生过程的设计理念、成长过程的思考、未来发展的期望。还对未来几年Java技术的发展情况和Java编程风格的现状和未来趋势阐述了自己的观点和期盼，让我们对未来几年如何提升和把握企业Java项目开发有了新的认识。



郑晔，ThoughtWorks公司首席咨询师，MOCO框架创始人。JavaOne 2013大会上，首届中国“Duke选择奖”得主之一。对Java和Ruby等方面的开发有着深刻的理解，致力于探寻合理的软件开发方式。07年加入ThoughtWorks公司后，投入到敏捷开发方法的实践之中，为其他公司提供敏捷开发方法方面的咨询服务。可以说MOCO的诞生，就是在ThoughtWorks创新传统下催生的。

以下是采访的具体内容：

记者：MOCO项目获得了今年Duke’s Choice Awards最具创新项目奖，能不能具体的说说MOCO框架在企业系统的集成测试方面有哪些针对性的优势?

嘉宾：可以说所有的团队在项目上都会遇到企业集成问题，但是却一直没有做出非常好的解决方案。我们知道，在测试的时候，我们需要一个模拟服务器。十几年前去做开发的时候没有现成的服务器，我们就从Socket最底层开始一层一层写到应用层，完成一个HTTP级的应用服务器。有过开发Java经验都知道，一次应用的重新部署，看似简单，但重复多次会影响整个团队项目的开发速度。所以就一直在想，能不能找到更好的解决方案，让开发人员在项目中省去这样的重复环节、让这个过程变得更轻松。这也是我直到现在都在追求的目标。

开始着手去做MOCO之前的一段时间，我翻译

了Martin Fowler的《特定语言》这本书，从书里获得了很多新的知识和想法。凭借十几年项目开发的经验总结，我决定开始自己动手解决这个问题。然后就有了MOCO这样一个解决方案，可以成功的解决企业集成问题。

记者：您刚才提到，在实际开发中很多团队都会遇到企业集成的问题，但却一直没有好的解决方案提出来。您认为导致这个现象的原因是什么？

嘉宾：目前来说，其实很多的项目开发团队都在尝试去解决这些问题，甚至有些公司还在自己的的开发团队里专门设立一个项目组去维护这样一个类似项目。但他们所做出来的框架仅仅用来解决自身项目，并没有想过把这样的框架泛化出来，作为一个通用的解决方案，应用到不同的项目中去。 所以，解决方案就停留在解决某个特定项目的层面上。

现在做技术，不应该仅仅是为了某个人或某个项目去做，而是应将技术成果分享出来，不断去完善和发散，使它成为一个通用技术方案，方便更多开发者使用。最近一段时间，公司一直在谈“创新漏斗”这样一个概念。“创新漏斗”倡导的就是在工作中发现问题，持续改进从而解决问题。然后，从中抽象出具有普适性的概念，再去积极地推广它。发现痛点、解决问题、泛化桥接、推广宣传。这也是MOCO的诞生之道。

记者：除了上述原因，从技术方面讲，它的实现有没有很高的技术要求在里面？抽象出来问题，但是在技术实现的层面会有一些难度？

嘉宾：有这种可能。其实整个软件开发过程中最难的事并不是代码的编写，而是能提供出一个问题的抽象模型。

在做MOCO解决方案的过程中，遇到的一些技术难点并新鲜。从十几年前做第一个项目开始就已经遇到过的，但一直都没有很好的解决办法。原因是无法将这些问题的概念抽象出来。经过了这些年的项目经验积累，脑子里才开始有了对这些问题的抽象概念。所以，直到去年才开发出来MOCO这样一个解决方案。

记者：有了新的功能以后，会不会把新的功能集成在MOCO框架里面，让它成为一个通用的功能？

嘉宾：会的。 MOCO去年已经基本完成了，但实际上直到现在MOCO框架的内容还在不断的修改和扩充。使用过MOCO框架开发团队，很及时的将他们在使用过程中遇到的问题和需求反馈给我，如果提出的需求符合MOCO框架的设计理念，就及时的将新的功能加到框架里,。只有这样MOCO才会不断地成长,这也是通用解决方案必须做到的一点。

记者：MOCO的发展是随着实际企业项目的需求而不断前进的，那么除了企业级的系统集成测试之外，它还具备了哪些较好的应用场景吗？

嘉宾： MOCO的设计目的旨在解决企业级项目系统集成问题。但实际上，MOCO的第一个应用，用在了移动开发项目里面。当时我们给成都的一家图书馆做一个项目，项目包括手机端和服务器端两个应用。一开始大家都想去做手机端的项目，而想去做服务器端的人寥寥无几。当时就在想，如果只有一个手机项目，我们怎么让他运转起来？后来的办法是，我们用MOCO搭了服务器，提供各种服务给手机应用。问题就这样解决了。也就在没有服务器端应用的情况下，把手机端实现了。后来服务器端的开发人员就参考MOCO的这套配置，比较轻松的将服务器端做了出来。■

本文未完，请阅读原文：
http://developer.51cto.com/art/201308/407194.htm

专访W3C中国区负责人李安琪：HTML5标准进展与最佳实践

□ 小林 /文



在很多Web开发者看来，HTML5之所以能够受到极热的追捧，它本身确有过人之处，与以往的技术相比，HTML5突出的优点非常明显。而在去年，由于HTML5的标准制定者WHATWG工作组和W3C最终选择了分道扬镳，给W3C组织很大的打击。但是，坚持照传统编号的版本控制系统构建一个静态快照版标准的W3C与致力于为HTML5制定一个不断进化的动态标准的WHAT工作组能够公开分手确实也在情理之中。

虽然说道不同不相为谋，作为激进的技术派WHATWG全力拖着保守组织W3C前进时，发现自己没办法保持双方步调一致，索性弃掉辎重轻装前进。但是，W3C前顾问Klaus Birkenbihl对眼下的局面却是非常的担忧，他认为：“W3C或者WHATWG都承受不了失败，因为这样将伤害到Web。”言下之意是，如果这两个组织都不能控制局面的话，那么市场就会失控。

然而，HTML5发展至今， W3C标准的被执行情况则不容乐观，根据MAMA工具分析，只有4.13%的网站完全符合W3C标准，另外50%的网站都在HTML中标示自己符合W3C标准，不过事实上很多网站可能在设计之初是符合规范的，但是因为内容变更或新的内容加入而不再符合标准。

在2013年17-18日由百度主办的Test The Web Forward（下面简称TTWF）在上海张江举行的活动，意图是让关注Web标准的开发者直接参与到Web平台的贡献中去，促进HTML5标准进入实践——此次TTWF测试用例征集活动将促进Web App标准的实现。51CTO记者被邀请前去，针对W3C标准的制定情况以及如何来加速标准制定和解决目前HTML5在浏览器中的兼容性等问题采访了W3C中国区总部负责人李安琪老师。

万维网联盟中国区负责人 李安琪

以下为相关的采访整理：

记者：关于标准化，大家都同意标准化的网络让普通的用户更方便，因为网络内容可以在不同地

方兼容，应该怎样来说服互联网产品公司来参与?

李安琪：宏观来说，标准化的Web带来更好的Web体验，覆盖更多的Web用户，创造更大的市场，提供更多的盈利空间及机会，使整个行业受益；可以把Web标准化的看成是做蛋糕，行业里的广大厂商一起努力把蛋糕做得更大，每个人才能分的更多。用一句W3C会员经常提到的话，a rising tide raises all boats。对于中国的ICT行业企业来说，虽然中国市场巨大，但也不可能与国际标准完全隔离。为了避免关键Web技术上受制于人的局面，积极的参与标准制定工作，表达自己的观点和利益诉求，有利于企业的长远发展战略。此外，一些国内的互联网企业已经发展壮大，并逐步摸索海外市场，标准化的技术有利于其产品进军海外市场，与国际同行同台竞争。

记者：我们都知道，最初HTML5想解决的就是兼容性问题，而在去年，HTML5标准制定组织WHATWG与W3C因为理念上的差异分裂。这就导致了会有两个版本的HTML5，这对HTML5的发展有什么影响?

李安琪：如果仔细看的话就会发现，WHATWG与W3C HTML工作组的最活跃的成员几乎都是同一批公司，他们是HTML5的主要推动者。WHATWG与W3C对行业有着不同的意义，WHATWG维护的living standard给行业提供一个相对激进的试验场（playfield），而W3C为行业提供一个成熟稳定且得到免费授权的标准版本；

记者：随着HTML5越来越受到开发者的喜爱，但是浏览器的碎片化问题开始显现，您认为这个问题将如何解决?

李安琪：HTML5标准的开发过程是应用驱动的过程。随着HTML5标准化进程在行业里的不断提高，浏览器的碎片化问题会逐渐得到改善；但是W3C所发布的标准，包括HTML5，都是推荐标准（recommendation），并非强制性标准，这些标准允许浏览器自主选择特性去实现，W3C尊重各厂商基于公司自身及市场情况所作出的关于产品的决定。浏览器厂商是HTMl5标准的主要推动者，相信他们也会是碎片化问题的主要解决者；我们呼吁开发者用标准的方式写应用，以适应不同的浏览器的能力，从开发人员的角度尽量避免应用所产生的碎片化问题。

记者：在HTML5进化过程中，不同浏览器支持的技术不同，我们看到不断分裂、又不断统一的情况，至今，严重的分裂仍然存在，例如db标准、多媒体格式等。 能否谈谈标准化方面的最新进展?

李安琪：HTML5.1最新标准进展请见8月18日的最新版本http://www.w3.org/html/wg/drafts/html/master/，最新扩展标准如下：

<http://dev.w3.org/html5/rdfa/>

HTML+RDFa – RDFa WG

<http://html5.org/specs/dom-parsing.html>

DOM Parsing and Serialization – Web Apps WG

<http://www.w3.org/TR/2012/WD-shadow-dom-20120522/>

Shadow DOM –Web Apps WG

<http://dvcs.w3.org/hg/web-intents/raw-file/tip/spec/Overview.html>

Web Intents – Web Apps WG / Device APIs WG

<http://dev.w3.org/html5/html-xhtml-author-guide/html-xhtml-authoring-guide.html>

Polyglot Markup: HTML-Compatible XHTML Documents – HTML WG

<http://dev.w3.org/html5/alt-techniques/>

HTML5: Techniques for providing useful text alternatives – HTML WG

<https://dvcs.w3.org/hg/editing/raw-file/tip/editing.html>

HTML Editing APIs – HTML Editing APIs CG

<http://www.w3.org/TR/html-media-capture/>

HTML Media Capture – Device APIs WG

<http://www.w3.org/TR/mediacapture-streams/>

Media Capture and Streams – Device APIs WG / WebRTC WG

<http://www.w3.org/TR/media-frags/>

Media Fragments URI – Media Fragments WG

<http://dvcs.w3.org/hg/html-media/raw-file/tip/encrypted-media/encrypted-media.html>

Encrypted Media Extensions – HTML WG

<http://dvcs.w3.org/hg/html-media/raw-file/tip/media-source/media-source.html>

Media Source Extensions – HTML WG

记者：目前业界对HTML5和CSS3等标准制定的进度迟缓抱怨很多，您能介绍一下这两个关键标准组织的进展情况吗？

李安琪：HTML5等标准里面很多特性牵动很多行业和产品，利益交锋比较激烈，深入的探索和讨论需要一定的时间；除此之外，保证标准的成熟稳定需要大量的测试工作，也需要很长的时间。目前HTML5的讨论主要围绕在内容保护的DRM等问题，最新进展见上面的扩展标准列表；CSS工作组最新进展包括：

<http://www.w3.org/TR/2013/CR-css-text-decor-3-20130801/>

CSS TextDecoration Module Level 3候选标准（Candidate Recommendation），

<http://www.w3.org/TR/2013/WD-css-cascade-3-20130730/>

CSS Cascading and InheritanceLevel 3，

更新了

<http://www.w3.org/TR/2013/CR-css3-values-20130730/>

CSS Values and Units Module Level 3候选标准（Candidate Recommendation）等。

记者：鉴于HTML5标准尚未最终落地，各家浏览器支持程度不一，您认为HTML5最佳实践领域在哪里？

李安琪：移动应用，Web App。

记者：目前W3C在做些什么样的工作，来让更多的人来同意和采用W3C的标准？

李安琪：如前面所述，标准的兼容的Web符合行业的整体利益，这个理念得到了行业的深度认同。W3C需要和行业一起积极识别那些亟需被标准化的技术，满足行业不断发展的需求，促进行业发展，这样才会有更多的人认同并采用W3C的标准。

记者：您觉得标准化在中国是什么情况？草根力量对加速HTML5早些落地帮助多大？

李安琪：Web标准化在国内处于起步阶段，但是中国企业对W3C标准制定工作的参与和贡献正在快速增加。一线的开发人员，即草根力量，是HTML5标准的具体使用者，广大工程师对HTML5的普及和推广具有至关重要的作用。我们希望通过更多的途径帮助开发人员了解、学习并使用HTML5，促进HTML5的早日落地。

移动设备上的HTML5

记者：HTML5技术的发展推动了移动浏览器的变革，包括百度手机浏览器以及其他各家浏览器几乎都对HTML5进行支持，但目前都是从跑分的高低来分析对HTML5支持度，您认为这个分数是否能决定它对HTML5的兼容性或者稳定性呢？

李安琪：各浏览器厂商进行跑分测试，解决了一些基本的兼容性问题，有积极的意义。但如果为了跑分而跑分，钻测试的空子，就失去了标准化和市场竞争的真正意义。

记者：我们都知道最近微信5.0更新，带来了一个“飞机大战”的游戏，而这个游戏就是一个HTML 5的网页游戏，目前看上去确实很粗糙，没法和FLASH游戏相比，但是“飞机大战”的火爆，或许成为一直以来深陷叫好不叫座的HTML 5游戏强力推动机器，您认为未来社交应用对HTML5的发展有什么样的作用？

李安琪：机遇：随着移动智能设备的广泛普及，越来越多的社交应用是基于Web的，天然适合Web app发展，这为HTML5提供了更好更广的应用平台，对HTML5有很大的促进作用；挑战：新的Web社交应用大量出现，现有的HTML5也许不能满足这些社交应用的全部需求，需要不断完善HTML5的性能；W3C十分重视Social Network行业对Web标准的需求，刚刚于8月7-8日在美国三藩市举行了Social Standards 研讨会【1】，去理解并识别社交网络行业对Web标准的需求。

【1】http://www.w3.org/2013/socialweb/

记者： W3C推出了一个移动标准检测工具：MobileOK Checker。现在大部分人认为移动互联网将会是互联网的未来。但是一些互联网企业，像浏览器企业，现在在担心移动平台的标准化问题。那么移动互联网的标准化现状如何？

李安琪：与移动互联网相关的Web标准制定根据技术领域不同分散在W3C的很多小组里积极进行，如DeviceAPI WG，CoreMobile CG等。■

51CTO视频课程推荐

购买李海园讲师会员卡：600金币看3000金币课程！



疯狂软件Hibernate视频教程



封面设计：@51CTO小林

《开发月刊》电子杂志

51CTO开发频道

